

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



**Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación**

TRABAJO FIN DE GRADO

**ADAPTACIÓN PERSONALIZADA EN INTERFACES
CEREBRO-MÁQUINA AÑADIENDO INFORMACIÓN DE
FRECUENCIA Y FASE**

**Juan Rico Maldonado
Tutor: Pablo Varona Martínez
Julio 2018**

ADAPTACIÓN PERSONALIZADA EN INTERFACES CEREBRO-MÁQUINA AÑADIENDO INFORMACIÓN DE FRECUENCIA Y FASE

AUTOR: Juan Rico Maldonado
TUTOR: Pablo Varona Martínez

Grupo de Neurocomputación Biológica
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2018

Resumen

En este Trabajo Fin de Grado se va a incorporar la información de la fase a una interfaz cerebro máquina basada en potenciales visuales evocados para comprobar que esta información contribuye a mejorar su rendimiento y eficacia. Las interfaces cerebro máquina son sistemas de comunicación entre una persona y un ordenador que generalmente son usadas con pacientes que sufren algún tipo de discapacidad motora que les impide comunicarse o realizar otro tipo de tareas. Mediante el uso de estos sistemas el paciente puede comunicarse tanto con una máquina como con otras personas.

Estos interfaces todavía no tienen un uso generalizado y se deben mejorar las tasas de transmisión de información para conseguir una comunicación fluida con el paciente. En este trabajo, el sistema estará basado en el uso de potenciales visuales evocados de estado estacionario (SSVEP, siglas en inglés). Para ello se presenta una serie de estímulos visuales simultáneos en forma de fuentes de luz parpadeando a distintas frecuencias. Cuando el paciente centra su atención visual en una de esas luces (la que codifica el comando que quiere realizar), se puede registrar mediante un electroencefalograma actividad cerebral a dicha frecuencia. Añadiendo a este sistema la información de la fase del estímulo visual, se espera mejorar la decodificación del estímulo en el electroencefalograma y que el sistema se adapte mejor a cada usuario.

El sistema que se ha diseñado utiliza el control de LEDs en una placa Arduino para producir la frecuencia y la fase de los estímulos parpadeantes. Se realizarán pruebas para comprobar la exactitud de la generación de estímulos a una frecuencia y fase determinadas y se realizarán pruebas de validación del sistema desarrollado en sujetos reales para comprobar que se puede utilizar adecuadamente la información de la fase en el electroencefalograma.

Abstract

In this Bachelor Thesis the information of the stimulus phase was added to a brain computer interface (BCI) based on visual evoked potentials to test that this information can improve its performance and overall efficiency. Brain computer interfaces are communication systems between a person and a computer, which are generally used with patients who suffer some kind of motor disability preventing them from communicating or realizing other type of tasks. With the help of these systems, the patient can communicate with a computer or with other people.

These interfaces have not been widespread yet, the information transmission ratio should be improved to achieve a fluid communication with the patient. In this work, the system will be based on the use of steady state visually evoked potentials (SSVEP). To that end, a series of visual stimulus are presented simultaneously, as lights flashing at different frequencies. When the patient focuses his/her visual attention on one of these lights (the one that encodes the command that the patient wants to realize), brain activity at this frequency can be recorded by an electroencephalogram. By adding the information of the phase of visual stimulus to the system we aim to improve the decoding and to adapt better to each user specificity.

The system that has been designed uses the control of LEDs on an Arduino board to produce the frequency and phase of the flickering stimulus. Several tests were carried out to test the accuracy of the stimulus generation with the frequency and phase desired.

Further validation tests were carried out with volunteers to verify that the information of the phase can be used to improve SSVEP detection from the electroencephalogram recording.

Palabras clave

Interfaz cerebro-ordenador, interfaz cerebro-máquina, BCI, potenciales visuales evocados de estado estacionario (SSVEP), fase, frecuencia, electroencefalograma (EEG), adaptación al usuario, Arduino.

Keywords

Brain-computer interface, brain-machine interface. BCI, steady state visually evoked potential (SSVEP), phase, frequency, electroencephalogram (EEG), user adaptation, Arduino.

Agradecimientos

A mi familia, a mis amigos, en especial a Mario y a Luis, y a mi tutor y a David por ayudarme a realizar este proyecto.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	1
1.3	Organización de la memoria.....	2
2	Estado del arte	3
2.1	Interfaces cerebro máquina.....	3
2.2	Métodos de adquisición de señales.....	4
2.2.1	Electroencefalografía (EEG):	5
2.2.2	Electrocorticografía (ECoG):	5
2.2.3	Magnetoencefalografía (MEG):	5
2.2.4	Intracortical Neuron Recording (INR):	5
2.2.5	Imagen por resonancia magnética funcional (fMRI):	6
2.2.6	Near Infrared Spectroscopy (NIRS):	6
2.3	Origen de las señales neurológicas.....	6
2.4	Señales de control de los sistemas BCI	6
2.4.1	SSVEP:	7
2.4.2	C-VEP.....	10
2.4.3	Potenciales evocados P300:.....	10
2.4.4	Potenciales corticales lentos:	10
2.4.5	Ritmos sensoriomotores:	10
3	Diseño y desarrollo.....	13
3.1	Método de registro de señales	13
3.2	Método de estimulación	13
3.3	Uso de información de frecuencia y fase	14
3.3.1	Uso de información de la frecuencia	14
3.3.2	Uso de información de la fase	14
3.4	Arduino.....	15
3.5	Sistemas de grabación	16
3.5.1	Sistema G-TEC.....	17
3.6	Procedimiento.....	18
3.6.1	Codificación de la fase	19
3.6.2	Detección de frecuencia y fase	21
4	Integración, pruebas y resultados	25
4.1	Validación de frecuencias y fases.....	25
4.2	Prueba fotodiodo	28
4.3	Pruebas en sujetos reales	34
5	Conclusiones y trabajo futuro.....	39
5.1	Conclusiones.....	39
5.2	Trabajo futuro	39
	Referencias	41
	Glosario	43
	Anexos.....	I
A	Manual del programador	I

INDICE DE FIGURAS

ILUSTRACIÓN 1: ESQUEMA DE LAS FASES DEL MODELO GENERAL DE UN SISTEMA BCI.	3
ILUSTRACIÓN 2: REPRESENTACIÓN DE UN SISTEMA BCI BASADO EN SSVEP, CON ESTÍMULOS PARPADEANTES.....	7
ILUSTRACIÓN 3: ESQUEMA DEL EXPERIMENTO DE LA BÚSQUEDA DE MEJORES FRECUENCIAS CON EL ALGORITMO DE CICLO CERRADO.	9
ILUSTRACIÓN 4: BCI CON SISTEMA DE DELETREO CON USO DE INFORMACIÓN DE LA FASE.	15
ILUSTRACIÓN 5: PLACA ARDUINO UNO.	16
ILUSTRACIÓN 6: POSICIÓN DE LOS ELECTRODOS DEL CASCO EMOTIV EPOC SEGÚN EL ESTÁNDAR 10-20.....	17
ILUSTRACIÓN 7: SISTEMA DE ADQUISICIÓN, AMPLIFICADOR G.USBAMP.	18
ILUSTRACIÓN 8: CASCO DE ELECTRODOS SECOS G.GAMMASYS Y DETALLE DE ELECTRODO SECO G.SAHARA.	18
ILUSTRACIÓN 9: ESQUEMA DE LA INTRODUCCIÓN DE POSIBLES MEJORAS DE UNA INTERFAZ BASADA EN SSVEP TANTO EN EL RENDIMIENTO COMO EN LA ADAPTACIÓN AL USUARIO DENTRO DEL CONTEXTO DE ESTE PROYECTO.	19
ILUSTRACIÓN 10: EJEMPLO SEÑALES DE LOS LED PARA FRECUENCIA DE 5Hz Y FASES 0 Y $\pi/2$	20
ILUSTRACIÓN 11: DESCOMPOSICIÓN DE UNA SEÑAL CUADRADA EN SEÑALES SINUSOIDALES.....	22
ILUSTRACIÓN 12: EXPLICACIÓN CCA.....	22
ILUSTRACIÓN 13: EJEMPLO DE SEÑALES DE ESTÍMULO GENERADAS CON DISTINTAS FASES (0, $\pi/2$, π , Y $3\pi/2$) Y FRECUENCIAS (7, 10, 15 Y 22 Hz).	26
ILUSTRACIÓN 14: FFT DE LAS SEÑALES DE GENERACIÓN DE LOS ESTÍMULOS.	26
ILUSTRACIÓN 15: EVALUACIÓN DEL <i>JITTER</i> TEMPORAL EN LA GENERACIÓN DE SEÑALES PARA DISTINTAS FRECUENCIAS Y FASES.	27
ILUSTRACIÓN 16: CIRCUITO DEL FOTODIODO Y CONECTOR BNC Y FOTO DEL DISPOSITIVO CONSTRUIDO.....	28
ILUSTRACIÓN 17: MEDIDAS DEL FOTODIODO DURANTE LA GENERACIÓN DE COMANDOS DE ENCENDIDO PARA UN SOLO LED A LA VEZ.....	29
ILUSTRACIÓN 18: MEDIDAS FOTODIODO PARA LEDs 1 Y 4.....	30
ILUSTRACIÓN 19: MEDIDAS FOTODIODO PARA LEDs 2 Y 5.....	30
ILUSTRACIÓN 20: MEDIDAS FOTODIODO PARA LEDs 3 Y 6.....	31

ILUSTRACIÓN 21: MEDIDAS LEDs JUNTO A MEDIDAS FOTODIODOS.	31
ILUSTRACIÓN 22: MEDIDAS DIFERENCIAS DE TIEMPO ENTRE SUBIDAS DE LA SEÑAL OBTENIDA DEL FOTODIODO.	32
ILUSTRACIÓN 23: MEDIDAS DE LATENCIA DE ENCENDIDO.	33
ILUSTRACIÓN 24: PERIODO SEÑAL 7Hz PWM Y SU MEDIA.	34
ILUSTRACIÓN 25: SEÑAL SINUSOIDAL GENERADA CON PWM Y LUZ EMITIDA.	34
ILUSTRACIÓN 26: EJEMPLO PRUEBAS DE ESTIMULACIÓN A SUJETOS REALES CON SISTEMA DE GRABACIÓN GTEC.	35
ILUSTRACIÓN 27: EJEMPLO SEÑAL EEG CON SSVEP Y SEÑAL DE ESTÍMULO.	35
ILUSTRACIÓN 28: RESULTADOS DE LAS FASES OBTENIDAS.	37

INDICE DE TABLAS

TABLA 1: DURACIONES PRIMER PERIODO DE LA SEÑAL MEDIDA POR EL FOTODIODO.	32
TABLA 2: MEDIAS DE LA LATENCIA DE ENCENDIDO.	33
TABLA 3: TASA DE DETECCIÓN Y MEJORES FRECUENCIAS PARA CADA USUARIO.	36

1 Introducción

1.1 Motivación

Desde su origen, los ordenadores han sido controlados típicamente por medio de dispositivos que requieren de la acción del sistema motor y los músculos de las manos, como el ratón y el teclado. Actualmente se pueden controlar por otros medios como pueden ser los sistemas de reconocimiento de voz (Hirschberg and Manning, 2015) y de gestos oculares (Rozado et al., 2012a, 2012b, 2015), que año tras año mejoran y hacen más fácil nuestra interacción con los ordenadores. Sin embargo, las personas que presentan algunos tipos de discapacidad motora grave no pueden comunicarse con los ordenadores a través del teclado o el ratón, y algunas de ellas ni siquiera con la voz o el seguimiento ocular.

Es aquí donde entran en juego las interfaces cerebro-máquina. Estos dispositivos utilizan señales cerebrales para enviar comandos a los ordenadores de la misma forma que hacemos con un teclado o un ratón, lo que posibilita su uso en personas con discapacidades motoras (Lebedev, 2014; Wolpaw et al., 2002). Estas señales cerebrales tienen diversos orígenes. En este proyecto nos centraremos en las señales producidas por estímulos parpadeantes, llamados potenciales visuales evocados (*steady state visually evoked potentials*, SSVEP), ya que estos estímulos pueden ser producidos de forma relativamente sencilla y las señales pueden ser recogidas mediante un casco de electroencefalografía (Fernandez-Vargas et al., 2013a; Herrmann, 2001).

El uso de las interfaces cerebro máquina todavía no está demasiado generalizado, ya que las señales cerebrales son de naturaleza muy diversa y son complicadas de clasificar y decodificar. Además, dependiendo del método usado para recoger las señales, estas pueden ser de baja calidad. Estos factores hacen que la fidelidad de estos sistemas sea un punto complicado, es imprescindible que el sistema decodifique las instrucciones del usuario con el mínimo error ya que un usuario discapacitado puede depender de la interfaz cerebro máquina para comunicarse o realizar tareas como desplazarse. El avance de las tecnologías está haciendo posible llevar a cabo estas tareas en tiempo real, aunque las tasas de transmisión son bastante bajas (Nicolas-Alonso and Gomez-Gil, 2012). La optimización y la adaptación personalizada de los interfaces cerebro-máquina puede expandir su uso más allá de la implementación de mecanismos de comunicación para las personas discapacitadas (Fernandez-Vargas et al., 2013b). La motivación de este proyecto es mejorar la diferenciación y la rapidez en la decodificación de estas señales favoreciendo a la vez que la estimulación sea más cómoda para el usuario. Para ello se hará uso de la información de la frecuencia y la fase de los SSVEP.

1.2 Objetivos

El objetivo de este proyecto es diseñar un dispositivo mediante la plataforma Arduino para añadir la información de la fase en una interfaz cerebro-máquina basada en potenciales visuales evocados con el objetivo de mejorar la tasa de transmisión desde el usuario a la computadora, ya que este es uno de los principales problemas a los que debe enfrentarse este tipo de sistemas antes de que se generalice su uso.

Para llevar a cabo este objetivo se realizará un estudio del estado del arte de la tecnología actual, se configurará una interfaz cerebro-máquina basada en SSVEPs para que haga uso de la mejor frecuencia para cada usuario y de la información de la fase a la hora de codificar y decodificar la señal generada por el sistema de control de los LEDs diseñado, y se procederá a realizar pruebas en sujetos para comprobar que esta información puede utilizarse para mejorar la tasa de transmisión.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Estado del arte:** en este capítulo se da una visión general sobre el estado actual de la tecnología referida a los sistemas BCI. Para ello se explica en qué consiste un BCI, los métodos de adquisición de señales cerebrales existentes y los tipos de señales de control de estos sistemas.
- **Diseño y desarrollo:** en este capítulo se explican las decisiones tomadas en cuanto a las opciones de hardware y software disponibles para implementar el dispositivo de estimulación desarrollado en este proyecto.
- **Integración pruebas y resultados:** en este capítulo se explican las pruebas realizadas y se muestran sus resultados para comprobar el buen funcionamiento de software y hardware utilizado en este proyecto así como las pruebas realizadas en sujetos.
- **Conclusiones y trabajo futuro:** en este último capítulo se expondrán las conclusiones extraídas de la realización de este trabajo y los posibles caminos a seguir para trabajos futuros.

2 Estado del arte

2.1 Interfaces cerebro máquina

Las **interfaces cerebro máquina** (BMI) o **interfaces cerebro ordenador** (BCI) son sistemas de comunicación software y hardware que permiten controlar ordenadores o dispositivos externos a través de actividad cerebral. Este tipo de interfaces crean un canal para transmitir un mensaje desde una persona a un dispositivo externo sin la necesidad de usar actividades motrices. Se trata, por tanto, de una alternativa con muchas posibilidades para personas con discapacidades motoras (Allison et al., 2007; Choi et al., 2017; Nicolas-Alonso and Gomez-Gil, 2012).

Los sistemas BCI surgieron como medio de comunicación para pacientes discapacitados, como por ejemplo con síndrome de enclaustramiento, derrame cerebral o esclerosis lateral amiotrófica. De ese modo se usan señales de la corteza cerebral para controlar dispositivos externos en pacientes con lesiones del sistema nervioso. Además de estos usos, está surgiendo un nuevo campo de investigación en torno al uso de estos sistemas para mejorar la rehabilitación neurológica en pacientes con enfermedades como el trastorno de déficit de atención (Carelli et al., 2017).

El modelo genérico de los sistemas BCI consta de cuatro fases:

Adquisición de señal: se registra la actividad cerebral del usuario con electrodos y se prepara la señal para su procesamiento, amplificándola y digitalizándola. Además, se almacena para permitir su posterior análisis.

Procesado y decodificación de señal: recibe la señal digitalizada y la convierte en comandos que entiende el dispositivo. En esta fase se elimina el ruido, se obtienen las características y se decodifica el comando que se quiere evocar.

Aplicación de los comandos: se reciben los comandos de control y se realizan las acciones correspondientes en el dispositivo, por ejemplo, a través de una interfaz para deletrear o controlar una silla de ruedas.

Configuración: permite al operador (que puede ser una persona o un algoritmo) definir los parámetros del sistema en función de los resultados obtenidos para optimizar el uso de la interfaz.

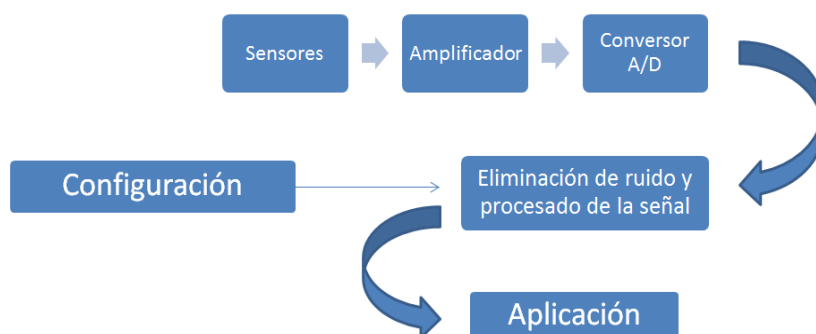


Ilustración 1: Esquema de las fases del modelo general de un sistema BCI. Adaptado de (Merchán, 2008).

Hay dos tipos de sistemas BCI, **invasivos** y no **invasivos**. Los sistemas no invasivos están basados en la técnica de electroencefalografía (EEG), que mediante electrodos en la superficie de la cabeza es capaz de registrar la actividad cerebral. Dan solución a personas con distintos tipos de parálisis para comunicaciones simples y evitan los riesgos de la cirugía cerebral aunque proveen una comunicación bastante limitada con tasas de transmisión y calidad de señal bajas. Por otro lado, los sistemas invasivos utilizan electrodos implantados intracranealmente. Así se consiguen señales de mucha mejor calidad, sin embargo el riesgo para los pacientes es muy alto, por lo que se emplean únicamente en pacientes con patologías muy graves (Lebedev and Nicolelis, 2006).

Este tipo de sistemas ha avanzado bastante en los últimos años. Sin embargo, hay distintos problemas a los que debe sobreponerse. Por ejemplo, para los sistemas BCI en los que el paciente imagina movimientos y así genera señales cerebrales que son decodificadas, un 20% de los pacientes no consiguen controlarlos de manera adecuada, fenómeno que se denomina '**BCI illiteracy**'. Se ha demostrado que otro tipo de sistemas que usan métodos como P300 o SSVEP (de los que más adelante hablaremos) funcionan para más cantidad de usuarios, aunque nunca para el cien por cien. Otro problema es que las señales cerebrales no son estacionarias, y en muchos casos requieren entrenamiento, ya que varían entre distintos sujetos y varían con el tiempo en un mismo sujeto. Para intentar solucionar estos problemas se ha propuesto un nuevo modelo de sistemas BCI híbridos que combinan sistemas BCI con otro sistema que utilizan señales neurológicas, fisiológicas (como la frecuencia cardiaca) o señales externas (Choi et al., 2017).

Los sistemas **BCI híbridos** se caracterizan por adquirir la actividad directamente del cerebro, usando al menos dos modalidades de adquisición como puede ser potenciales eléctricos o campos magnéticos, y por que la señal debe ser procesada en tiempo real y debe haber feedback al usuario para mejorar la comunicación entre el usuario y la máquina. En los sistemas BCHI (BCI híbridos) se buscan tres objetivos; mejorar la precisión de los BCI combinando señales de diferentes dispositivos de la misma actividad cerebral, aumentar el número de comandos de control de la aplicación sin disminuir la precisión del sistema decodificando actividades cerebrales de diferentes regiones con distintos sistemas, y por último, reducir el tiempo de detección de los comandos (Hong and Khan, 2017).

En cuanto a las perspectivas de futuro de los sistemas BCI, se esta buscando combinar estos sistemas con la fabricación de interconexiones sintéticas de neuronas. Actualmente aun no se ha conseguido la cura de tejido nervioso, por lo que se están investigando entornos para llegar a entender los mecanismos de crecimiento de las neuronas para desarrollar métodos de regeneracion neuronal (Marcus et al., 2017).

2.2 Métodos de adquisición de señales

Los métodos más habituales para adquirir las señales cerebrales son los siguientes:

2.2.1 Electroencefalografía (EEG):

Consiste en una técnica de exploración mediante la cual se registra la actividad eléctrica cerebral en tiempo real. El origen de la señal está en la corteza cerebral. Para recoger una señal de la actividad eléctrica a través de la superficie craneal se colocan electrodos que captan la diferencia de potencial entre ellos. Se trata de una técnica indolora, no invasiva y de bajo coste. Hay dos tipos de montaje, el bipolar, que registra la diferencia de voltaje entre dos electrodos colocados en áreas de actividad cerebral y el monopolar, que registra la diferencia de potencial entre un electrodo ubicado en una zona cerebral activa y otro colocado en una zona sin actividad (Ramos-Argüelles et al., 2009).

A la hora de realizar el EEG se debe tener en cuenta la presencia de artefactos, es decir, de variaciones no deseadas recogida en los datos del EEG (como pueden ser los artefactos debidos al parpadeo) que pueden llevar a la corrupción de los datos del EEG o a la mala interpretación de los comandos del usuario por parte del sistema BCI. Varios estudios han demostrado que eliminar las señales procedentes del parpadeo mejora el rendimiento de los sistemas BCI (Hong and Khan, 2017).

2.2.2 Electrocuticografía (ECoG):

En este caso, los electrodos se colocan directamente en la superficie del cerebro después de realizar una craneotomía por lo que solo se usa en casos de extrema gravedad debido a los riesgos que implica la cirugía. La señal extraída tiene mayor resolución y es menos vulnerable a parpadeos o movimientos de ojos que la obtenida de un EEG (Nicolas-Alonso and Gomez-Gil, 2012).

2.2.3 Magnetoencefalografía (MEG):

Se trata de una herramienta que permite captar los campos magnéticos generados por los potenciales postsinápticos ocurridos en las dendritas de las neuronas. La ventaja de este método es que la señal no se degrada por el paso a través de los tejidos como ocurre con las señales captadas por el electroencefalograma (Maestú et al., 2005). El inconveniente es que el hardware necesario es más voluminoso y caro que el de EEG.

2.2.4 Intracortical Neuron Recording (INR):

Es un método invasivo que consiste en un *array* de electrodos implantados en la corteza del cerebro. Estos electrodos recogen la señal de grupos de neuronas en zonas muy específicas de la corteza cerebral con la resolución necesaria para el control de prótesis externas (Gunasekera et al., 2015).

2.2.5 Imagen por resonancia magnética funcional (fMRI):

Técnica no invasiva que detecta cambios en el volumen de sangre y los niveles de oxigenación durante la actividad neuronal por medio de campos electromagnéticos con una muy alta resolución espacial. La desventaja de este método es que tiene una resolución temporal baja, de 1 o 2 segundos. Además la respuesta hemodinámica es lenta, ya que los cambios vasculares ocurren varios segundos después de la actividad neuronal (Nicolas-Alonso and Gomez-Gil, 2012).

2.2.6 Near Infrared Spectroscopy (NIRS):

Técnica no invasiva que utiliza luz infrarroja para detectar los cambios de concentración hemodinámica. Al igual que el anterior, el retardo en la respuesta vascular hace que sea difícil implementar un BCI que reaccione de forma rápida (Zephaniah and Kim, 2014).

2.3 Origen de las señales neurológicas

Las señales neurológicas se pueden clasificar según su origen en **exógenas** o **endógenas**.

Las señales **exógenas** dependen de actividad neuronal evocada por estímulos externos, por lo que los sistemas BCI exógenos no necesitan en muchos casos entrenamiento. Estos estímulos son los llamados potenciales visuales evocados de estado estacionario (SSVEP) y los potenciales evocados P300 de los que hablaremos a continuación. Con estos métodos las señales pueden ser detectadas con un EEG con una tasa de información de 60 bits/min.

Las señales **endógenas** en cambio no dependen de estímulos externos. El usuario debe aprender a producir los patrones que son decodificados por el sistema. La duración del entrenamiento depende del sujeto y de la estrategia seguida en el experimento (Chan et al., 2015).

Las interfaces cerebro-máquina también se suelen clasificar como exógenas o endógenas dependiendo de si utilizan estimulación externa o no para evocar la actividad cerebral.

2.4 Señales de control de los sistemas BCI

Las señales cerebrales se componen de numerosos fenómenos en los cuales, en algunos casos, todavía no se conoce ni su origen ni su funcionamiento. En cambio, en otros casos, estos fenómenos fisiológicos han sido investigados y los usuarios pueden aprender a utilizarlos para enviar instrucciones al sistema BCI (Nicolas-Alonso and Gomez-Gil,

2012). Existen numerosas señales que se utilizan para controlar un sistema BCI, entre las que destacan las que se comentan en las siguientes secciones:

2.4.1 SSVEP:

Los sistemas BCI basados en **SSVEP** permiten al usuario elegir entre distintos comandos asociados con un estímulo visual repetitivo que tiene propiedades distintivas ya sean de frecuencia o fase. Los estímulos son presentados al usuario de forma simultánea y este debe centrar su atención en el estímulo que corresponde al comando que desea realizar. De esta forma, un SSVEP es provocado y se manifiesta en componentes oscilatorios en el EEG del usuario a las frecuencias del estímulo o sus armónicos (Zhu et al., 2010). Estos SSVEP pueden ser provocados por estímulos visuales con frecuencias de 1 a 100 Hz (Herrmann, 2001; Nicolas-Alonso and Gomez-Gil, 2012).

En la siguiente imagen se puede apreciar un ejemplo de cómo funcionan realmente los SSVEP. Se muestra el estímulo al usuario, en este caso, por ejemplo, un cuadrado parpadeando a una frecuencia determinada, 20 Hz. Al mismo tiempo se registra la actividad con un casco de electrodos. De ese modo, en el EEG resultante se obtiene una respuesta como la de la imagen, con un pico de amplitud a la frecuencia (o los armónicos de dicha frecuencia) a la que parpadea el estímulo donde el usuario está fijando su atención visual. Así, si se presentan varios estímulos en la pantalla, asociados a distintos comandos, parpadeando a diferentes frecuencias, si el usuario centra su atención en uno de esos estímulos, se puede decodificar el comando que el usuario desea realizar.

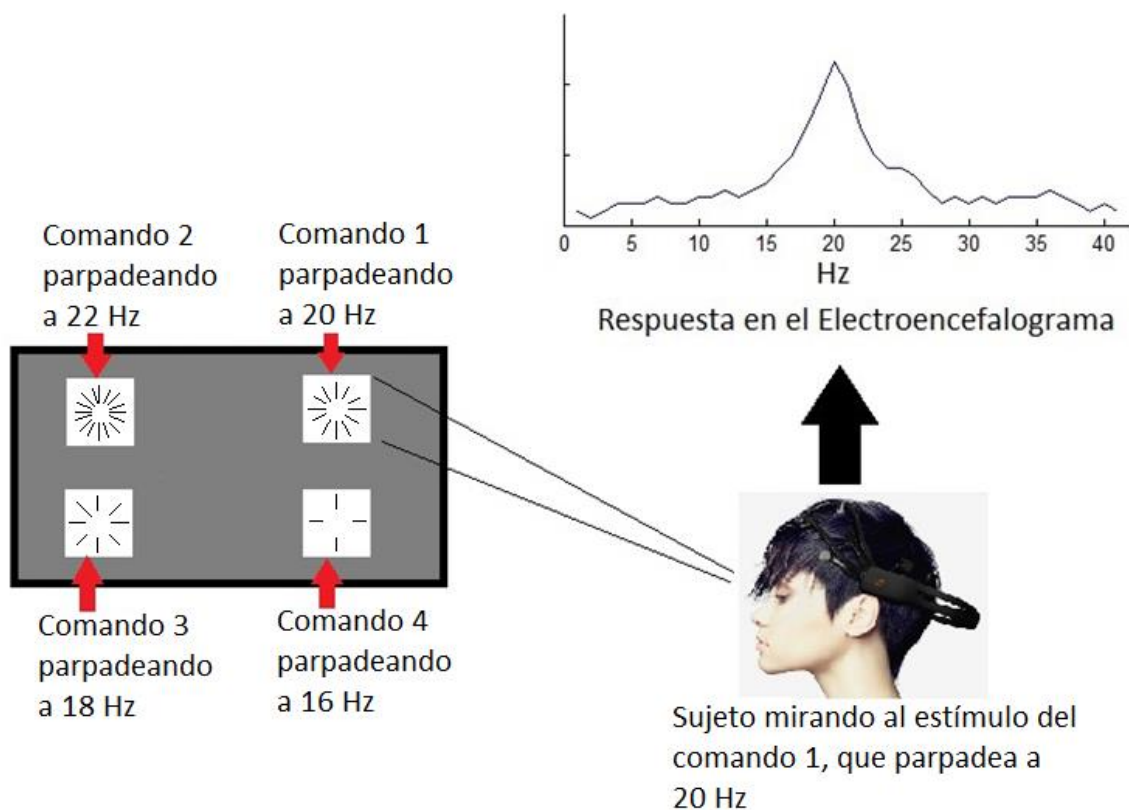


Ilustración 2: Representación de un sistema BCI basado en SSVEP, con estímulos parpadeantes. Adaptado de (Fernandez-Vargas et al., 2013a)

Los estímulos se clasifican en tres tipos, de luz, gráficos y con patrones inversos. Los estímulos de luz usan fuentes de luz como LEDs modulados a una frecuencia específica. Los estímulos gráficos son representados en una pantalla y parpadean a una frecuencia determinada. Este tipo de estímulos pueden ser, por ejemplo, un cuadrado blanco parpadeando sobre un fondo negro. En este caso se debe tener en cuenta la frecuencia de refresco de la pantalla en la que se representa el gráfico y por tanto elegir las frecuencias adecuadamente. Los estímulos con patrones inversos son representados también en una pantalla, pero en este caso consisten en patrones que cambian un número específico de veces por segundo. Estos patrones suelen ser tableros de ajedrez o simplemente líneas, de colores blanco y negro en las que las partes blancas pasan a negro y viceversa. Hay que tener en cuenta que, en el caso de los estímulos gráficos, provocan los SSVEP a la frecuencia de un ciclo completo, es decir, cuando para el caso de un cuadrado blanco sobre fondo negro, el cuadrado desaparece y vuelve a aparecer. Sin embargo, los estímulos con patrones inversos provocan SSVEP a la frecuencia de una sola alternancia, es decir, si se trata de un tablero de ajedrez, cuando los cuadrados negros pasan a blanco y vuelven a negro sería percibido como 2 ciclos (Zhu et al., 2010).

Para que el sistema BCI se comporte de forma óptima es necesario tener en cuenta las características de los estímulos, es decir, el tamaño, la forma, la distancia al estímulo, el color y la presencia de un punto de fijación en el centro de cada estímulo parpadeante. No existe un acuerdo general sobre qué color es el más adecuado. Los estudios realizados indican que el blanco, el amarillo y el verde son los que provocan respuestas más fuertes, mientras que el azul provoca respuestas más débiles (Duszyk et al., 2014) aunque el azul es menos sensible a la frecuencia usada, así como el amarillo (Zhu et al., 2010) y que el color violeta da mejores resultados que el verde, el rojo y el azul (Singla et al., 2013). Se puede asumir que los colores más oscuros provocan respuestas más débiles en comparación con colores brillantes debido al contraste entre el estímulo y el fondo de color negro (Duszyk et al., 2014). A la hora de medir el tamaño más eficaz, si se ha demostrado que los estímulos de mayor tamaño provocan respuestas de mayor amplitud. Esto se explica asumiendo que un estímulo más grande activa también una región más grande de la corteza visual. La distancia al estímulo y la forma resultaron no tener influencia en el experimento de (Duszyk et al., 2014). Por último, la presencia de un punto de fijación bajó la calidad de la respuesta en dicho experimento, sin embargo, esta disminución no era significativa.

El factor ambiente también es importante, se ha demostrado que una sala a oscuras tiene efectos positivos en la fuerza de la señal en los SSVEP. Además, los estímulos de mayor intensidad provocan una respuesta mayor en el usuario, sin embargo, también provocan un mayor cansancio y por tanto un menor grado de atención hacia dichos estímulos.

La respuesta de los SSVEP también depende en gran medida de la frecuencia de estimulación y de la compatibilidad fisiológica entre las frecuencias de todos los estímulos que se generan simultáneamente. En el experimento de (Fernandez-Vargas et al., 2013a) se realizó una búsqueda de ciclo cerrado para determinar el conjunto de frecuencias más compatible con cada individuo que realizó el experimento ayudado por retroalimentación auditiva cuando se alcanzaba el SNR objetivo y se comparó el comportamiento del sistema BCI para ese conjunto de frecuencias, para un conjunto de frecuencias prefijadas (a 27, 28, 29 y 30 Hz) y para un conjunto de frecuencias “top” con los mejores resultados para cada individuo. El estudio reveló que el protocolo de ciclo cerrado mejoraba la

eficiencia del BCI ya que proveía al sujeto y al sistema con información online para alcanzar el objetivo del BCI.

Gracias al estudio de todos estos factores en los sistemas basados en SSVEP se ha conseguido que estos sean los que proporcionan mayores tasas de transferencia hasta la fecha, llegando a los 4.45 bps mientras que otros métodos como P300 que se explicara a continuación, no pasa de 1 bps (Chen, et al., 2015).

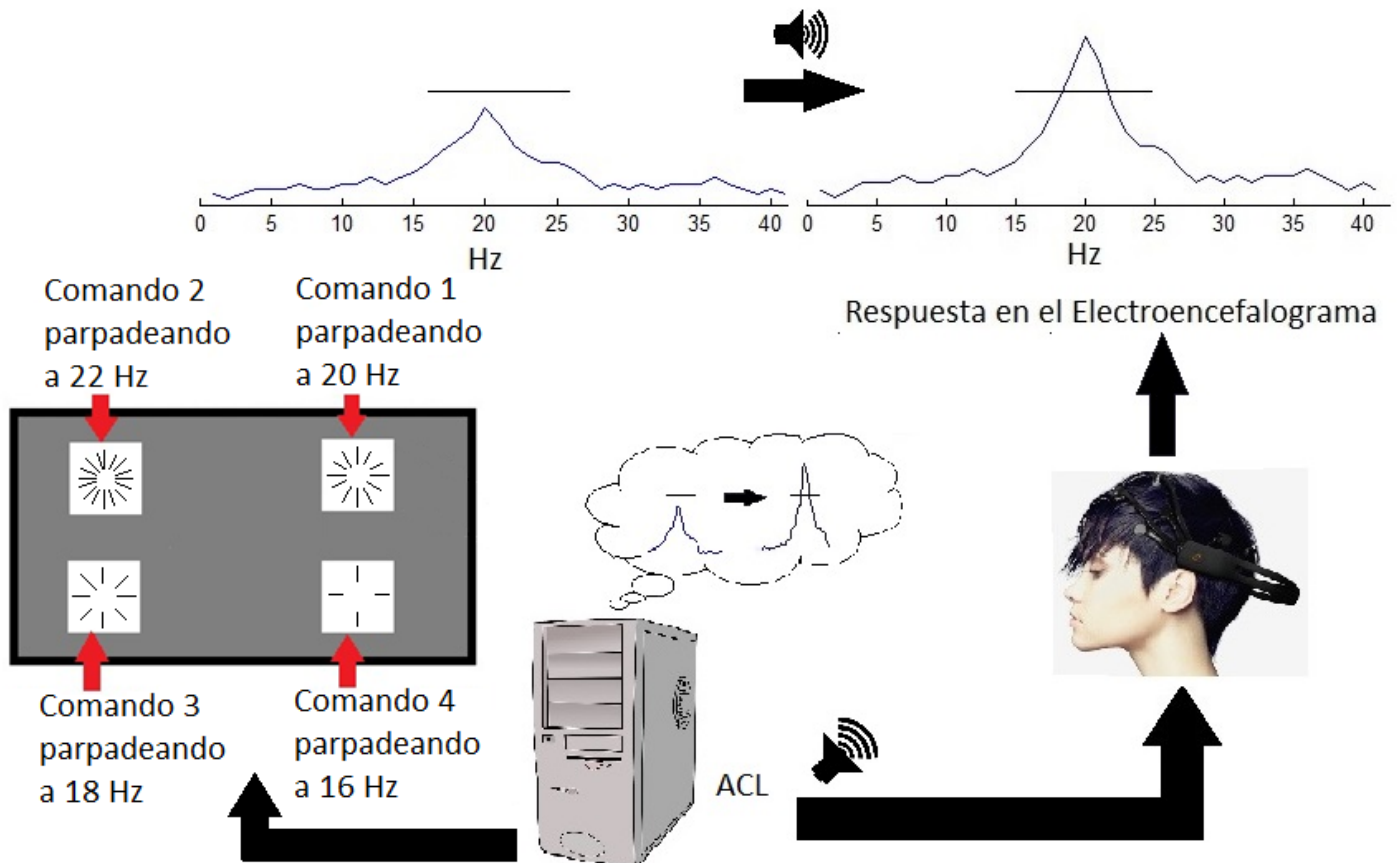


Ilustración 3: Esquema del experimento de (Fernandez-Vargas et al., 2013a) de la búsqueda de mejores frecuencias con el algoritmo de ciclo cerrado.

Por último, la información de la fase también puede suponer una mejora en el rendimiento con este tipo de estímulos. El estudio sobre este tema (Chen et al., 2015) describe un BCI para deletrear que consigue un incremento de la tasa de transmisión basado en la codificación de frecuencia y fase en SSVEP. El sistema resultante obtuvo tasas de hasta 60 caracteres por minuto. Por tanto, añadiendo la información de la fase se consigue mejorar la diferenciación entre SSVEPs con un rango de frecuencias mucho más estrecho.

2.4.2 C-VEP

Otro tipo de estímulos VEP (visual evoked potential) son los c-VEP o code modulated VEP. Este tipo de estímulo es generado igual que un SSVEP pero siguiendo una secuencia aleatoria, es decir, una luz o una imagen parpadeando en vez de a una frecuencia determinada, siguiendo una secuencia aleatoria alternando ciclos de encendido y apagado (Nicolas-Alonso and Gomez-Gil, 2012).

Mediante el uso de este tipo de estímulo se han llegado a conseguir tasas de transferencia bastante altas, de hasta 127.8 bits/min en el experimento de (Wittevrongel et al., 2017).

2.4.3 Potenciales evocados P300:

Este tipo de señales se generan por un estímulo visual, auditivo o somatosensorial, infrecuente o significativo, en el contexto de otros estímulos frecuentes, lo que evoca deflexión en el EEG 300 ms después de que suceda dicho estímulo (Cossio and Gentiletti, 2008). El uso de este tipo de BCI no requiere entrenamiento a priori, pero su rendimiento puede empeorar debido a que el usuario puede acabar acostumbrándose a los estímulos infrecuentes, lo que reduce la amplitud de los P300. Estos sistemas suelen consistir en una matriz de símbolos o comandos en la cual las filas y columnas se iluminan. El usuario mira fijamente el símbolo deseado y cuenta las veces que se ilumina el símbolo para no perder la concentración, así cada vez que la imagen se ilumina se provoca la respuesta P300 (Nicolas-Alonso and Gomez-Gil, 2012).

2.4.4 Potenciales corticales lentos:

Se tratan de variaciones lentas en los EEG de uno o varios segundos de duración de señales por debajo de 1 Hz. Se asocian a cambios en la actividad cortical. El control de este tipo de señales requiere entrenamiento, en algunos casos de meses, y depende de la habilidad de aprendizaje del usuario y su estado de ánimo. Además, las tasas de transferencia de información son bajas. Se suele usar en pacientes que están en las primeras etapas de enfermedades degenerativas como el ALS (Nicolas-Alonso and Gomez-Gil, 2012).

2.4.5 Ritmos sensoriomotores:

Este tipo de señales son usadas en los sistemas BCI ya que se puede aprender a controlarlas voluntariamente. Se trata de variaciones en la amplitud de los ritmos sensoriomotores que ocurren cuando hay actividad cerebral causada por movimientos motores aunque estos no terminen de llevarse a cabo, por ejemplo, se podrían producir estas variaciones al pensar en mover una extremidad sin que hiciera falta llegar a moverla, aunque para controlar BCI basados en ellos también es necesario el entrenamiento. Hay dos tipos de variaciones de amplitud en los ritmos sensoriomotores, event-related

desynchronization (ERD), que conlleva una disminución de la amplitud que comienza varios segundos antes que el propio movimiento y event-related synchronization (ERS) que al contrario que el anterior, supone un aumento en la amplitud (Nicolas-Alonso and Gomez-Gil, 2012).

En este proyecto se va a realizar un sistema BCI basado en SSVEPs en el que se utilizará la información de la frecuencia y la fase. Con ello se busca que la codificación y decodificación de las señales utilizadas sea más rápida para poder así aumentar la tasa de transferencia de información y una mayor adaptación al usuario.

3 Diseño y desarrollo

En esta sección se tratará cuáles son las opciones posibles para llevar a cabo este proyecto y los motivos por los que se han elegido.

3.1 Método de registro de señales

El primer aspecto a tener en cuenta es qué método se utilizará para recoger las señales cerebrales. De entre las opciones posibles, los métodos invasivos quedan descartados debido a su alto riesgo y a que requieren un estricto control hospitalario, por lo que quedan las opciones de electroencefalografía, magnetoencefalografía, imagen por resonancia magnética funcional y near infrared spectroscopy. Actualmente las máquinas utilizadas para la adquisición de señales basadas en resonancias magnéticas son demasiado aparatosas por lo que resulta inviable utilizar estos métodos para este proyecto. Por ello, el método que seleccionamos en este trabajo para la adquisición de señales es el EEG, en particular en sistemas de adquisición que utilizan electrodos de fácil colocación.

3.2 Método de estimulación

También se debe tener en cuenta qué tipo de estimulación se utilizará para decodificar los comandos que el usuario de la interfaz cerebro-máquina quiere producir. Puesto que los SSVEPs constituyen la alternativa más eficaz y rápida en el contexto de la tecnología actual de interfaces cerebro-máquina, se ha elegido este tipo de estimulación exógena. Como hemos visto anteriormente los estímulos visuales parpadeantes se pueden mostrar por una pantalla o se pueden utilizar LEDs. A priori, lo más intuitivo resultaría utilizar una pantalla de ordenador como interfaz en la que se mostrarán los estímulos, sin embargo, si se pretende utilizar un monitor surge el siguiente problema. La tasa de refresco de un monitor suele ser de 60 o 75 Hz. Por esta razón solo se pueden utilizar un número determinado de frecuencias que sean submúltiplos de estas. Además, entre ellas se debe tener en cuenta que hay armónicos y subarmónicos que se deben eliminar al no poder discernir entre ellas. Por último, se les debe poder aplicar una FFT, si usamos por ejemplo una ventana de 2s, la precisión de la transformada es de 0.5Hz, lo que reduce las frecuencias utilizables a un número muy reducido (Vargas, 2013). Asimismo, las frecuencias óptimas varían para cada sujeto, por lo que el conjunto de frecuencias utilizables resultante de usar un monitor no es lo más adecuado en el caso de querer realizar un sistema que se adapte al usuario. Además, a la hora de codificar comandos con diferentes frecuencias, cuantas más frecuencias se puedan utilizar, más comandos se podrán codificar.

La segunda opción para producir los estímulos es utilizar un circuito con LEDs controlado por un microcontrolador como puede ser una placa Arduino. De este modo se pueden utilizar todas las frecuencias que sean convenientes al no depender de la frecuencia de refresco del monitor, así como incluir la fase en la modulación de la señal, lo que

permitirá una mayor adaptación a las necesidades del usuario y el uso de un número mayor de estímulos y por tanto de comandos. Por tanto, la opción escogida para producir los estímulos será esta. Arduino también proporciona opciones para cambiar el brillo de los LEDs y es una herramienta de hardware libre con su propio entorno de desarrollo.

3.3 Uso de información de frecuencia y fase

Se busca que la eficacia del sistema BCI desarrollado sea la máxima posible y se adapte al usuario, por lo que se configurará el sistema BCI para elegir las mejores frecuencias para cada usuario. Esta adaptación se realizará también teniendo en cuenta la fase de la estimulación para codificar más efectivamente los comandos que el usuario quiere generar.

3.3.1 Uso de información de la frecuencia

Ya se demostró (Fernandez-Vargas et al., 2013a) que un protocolo de lazo cerrado que proporcione información online para seleccionar las mejores frecuencias en cada sujeto aumenta la eficacia del sistema. Normalmente en los sistemas BCI basados en SSVEP se utilizan frecuencias prefijadas, sin embargo, en dicho estudio se compararon los resultados de tasa de transmisión de información para sistemas usando frecuencias prefijadas y usando frecuencias adaptadas a cada usuario. Para seleccionar las mejores frecuencias para cada sujeto en ciclo cerrado, se exploraba la compatibilidad de las frecuencias de dos en dos y de cuatro en cuatro mediante una interacción cíclica. En cada iteración el usuario debía fijar su atención visual en una serie de estímulos parpadeantes y se calculaba la relación señal ruido de la respuesta para obtener un ranking de frecuencias. Además, para proporcionar información al usuario se utilizó un sistema de realimentación sonoro. De ese modo, cuando la señal registrada del usuario alcanzaba un umbral preestablecido y se mantenía por encima de ese umbral un tiempo determinado se pasaba al siguiente estímulo. Por tanto, en este experimento había dos ciclos cerrados, el primero para encontrar las frecuencias más compatibles y así mejorar las tasas de información, y el segundo para informar al usuario y así ayudarlo a alcanzar el objetivo de actividad cerebral necesario para detectar el comando que desea enviar al ordenador. En este artículo se demuestra que si se utilizan frecuencias adaptadas a cada sujeto se mejora el rendimiento del sistema, y en general que cada individuo tiene un conjunto de frecuencias óptimas.

3.3.2 Uso de información de la fase

En este TFG se hará uso también de la información de la fase en la estimulación, ya que se ha demostrado que aumenta la tasa de transferencia de información del sistema (Chen et al., 2015). En el citado experimento llevado a cabo usando información de la fase en estimulación evocada en un monitor se llegó a alcanzar tasas de 4.45 bps, es decir unas 12 palabras por minuto, mientras que normalmente otros sistemas BCI solo alcanzan de 5 a 10 palabras por minuto. En este experimento se diseñó una matriz de 5 por 8, con 40 caracteres, cada uno de ellos codificado parpadeando con una frecuencia y una fase

determinadas. Para añadir la información de la fase se implementó un sistema síncrono de modulación y demodulación. De esta manera, para seleccionar un comando (en este caso un carácter) se mantenía medio segundo sin estimular, después se estimulaba a la frecuencia y fases seleccionadas durante otro medio segundo. Así se producía un SSVEP con una latencia determinada, y se podía demodular la señal y detectar el comando seleccionado como se aprecia en la ilustración 4. El usuario centra su atención por ejemplo en la letra “P”, que parpadea a una frecuencia de 15Hz y tiene una fase de 0.25π . De ese modo, en el electroencefalograma se detecta una señal con un pico en 15 Hz con una fase de 0.25π , pudiendo decodificar esa señal como la letra “P”.

Letra "P" parpadeando a 15 Hz con fase 0.25π

Resto de letras parpadeando a distintas frecuencias con distinta fase

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	1	2	3	4	5	6



Usuario fijando la mirada en la letra "P"

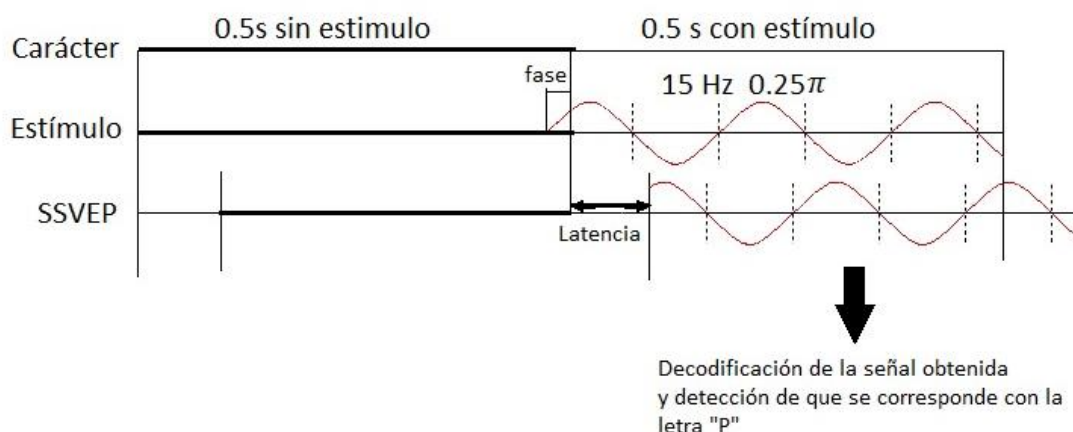


Ilustración 4: BCI con sistema de deletreo con uso de información de la fase. Adaptado de (Chen et al., 2015).

3.4 Arduino

Para llevar a cabo esta configuración en la que se desea utilizar información de la frecuencia y la fase en el estímulo, se ha decidido que lo más apropiado es utilizar un montaje con LEDs y una placa Arduino, ya que permite implementar el sistema con más precisión que usando una pantalla de ordenador.

Arduino es una de las plataformas de software y hardware libre más importante del mundo con su propio entorno de desarrollo y un lenguaje bastante similar a C++. Además, permite el uso tanto de Windows como Linux y Mac y tiene un coste relativamente asequible. Existen numerosas opciones a la hora de utilizar Arduino, entre las que consideraremos están las placas Arduino Yun y Arduino Uno, ya que son las placas de las que dispone el GNB (grupo de Neurocomputación Biológica).

Arduino Yun es una placa microcontroladora con distribución Linux, soporte Ethernet y WiFi, un puerto USB, una ranura micro SD y 20 pines de salida y entrada. Su principal característica es que permite el uso de Wifi. Arduino Uno consta de 14 pines digitales de entrada y salida, 6 entradas analógicas, una conexión USB y un reloj de 16 MHz de velocidad. Este proyecto parte del trabajo de fin de máster de Ana Sotomayor Romillo “Control adaptado al sujeto de una interfaz cerebro-máquina con señalización mixta” en el que se consideraba únicamente la generación de frecuencias de estímulo en placas Arduino. En dicho trabajo se implementó el sistema usando Arduino Uno, por lo que será la opción elegida para este proyecto.



Ilustración 5: Placa Arduino UNO.

3.5 Sistemas de grabación

Para realizar la electroencefalografía el laboratorio del GNB dispone de dos alternativas. El casco EMOTIV EPOC y el sistema de grabación G-TEC con electrodos secos g.SAHARA y el amplificador g.USBamp.

El casco EMOTIV EPOC se trata de un dispositivo inalámbrico, con sensores que no necesitan el uso de geles para recoger la señal. Tiene un total de 14 canales, es decir, 14 sensores, colocados en las posiciones AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 y AF4 según el estándar 10-20. Este estándar es el sistema internacional usado para colocar los electrodos en las pruebas de electroencefalografía. Cada posición en la que se colocan los electrodos viene designada por un número y una letra. Las letras identifican la zona de la cabeza, es decir, A para el lóbulo de la oreja, F para frontal, T para temporal, C para central y P para parietal (las combinaciones de letras se usan para posiciones intermedias), los números pares se utilizan para el hemisferio derecho y los impares para el hemisferio izquierdo (Wikipedia contributors, 2018). De tal modo que la situación de los electrodos del casco EMOTIV EPOC es la siguiente:

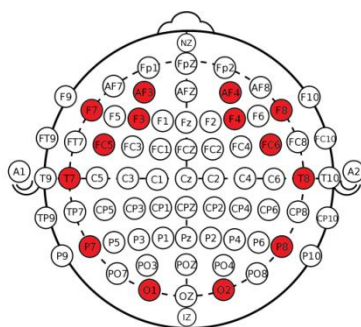


Ilustración 6: Posición de los electrodos del casco EMOTIV EPOC según el estándar 10-20.

Sin embargo, sus electrodos están en una posición fija, con solo dos en posiciones occipitales, que son las que suelen obtener mejores resultados en cuanto a la detección de la actividad cerebral de los SSVEPs. Por tanto, el sistema de grabación utilizado en este proyecto será el de g.TEC.

3.5.1 Sistema G-TEC

En cuanto al amplificador g.USBamp, se trata de un sistema de adquisición, amplificación y procesamiento de bioseñales, diseñado para la investigación. Su uso no solo se limita a EEG sino que también se puede utilizar para registrar actividad muscular, cardiovascular y respiratoria entre otros parámetros fisiológicos. Este sistema cuenta con 16 canales de 24 bits, garantizando que no haya interferencia entre las señales registradas, un rango de entrada de voltaje de 250 mV y una tasa de muestreo de hasta 34.8 KHz, pudiendo conectarse al ordenador mediante un puerto USB. Junto con este amplificador se deben utilizar electrodos, que pueden ser secos o húmedos. Los electrodos húmedos, aunque proporcionan mejor integridad de señal, suelen resultar más incómodos para el usuario ya que es necesaria la aplicación de geles en el cuero cabelludo, por lo que el usuario debe lavar tanto el casco como su propio pelo después de utilizarlos. Por esta razón se opta por el casco de electrodos secos de g.TEC, g.GAMMAsys. Los electrodos secos g.SAHARA son capaces de capturar un rango de frecuencias de 0.1 a 40 Hz, cubriendo así el rango en el que mejor se detectan los SSVEPs. Estos electrodos cuentan con ocho pines de aleación de oro de la longitud necesaria para llegar a estar en contacto con la piel del individuo a pesar del pelo. La gran ventaja de este casco respecto al EMOTIV EPOC es que el casco de G-TEC permite cambiar las posiciones de los electrodos y ponerlos en cada una de las posiciones del sistema 10-20 explicado anteriormente, mientras que los electrodos del EMOTIV EPOC son fijos. Por esta razón se usará el sistema G-TEC basándose en los resultados obtenidos en el trabajo de fin de grado de David Rodríguez Muñoz “Calibración automática y elección de las mejores referencias en interfaces cerebro máquina” para colocar los electrodos en las posiciones que proporcionen mejores resultados en cuanto a calidad de la señal. Por ello, a la hora de realizar las pruebas, los electrodos se colocaron en las posiciones FPZ, FP1, FP2, T7, T8, PZ, P3, P4, POZ, PO3, PO4, PO7, PO8, O1, O2 y OZ. La desventaja de este sistema de electrodos secos es que recogen también ruido externo y son muy susceptibles al movimiento del usuario.



Ilustración 7: Sistema de adquisición, amplificador g.USBamp.



Ilustración 8: Casco de electrodos secos g.GAMMAsys y detalle de electrodo seco g.SAHARA.

3.6 Procedimiento

Debido a todo lo explicado, este trabajo de fin de grado seguirá el siguiente esquema. El sistema se basará en SSVEP generados por estimulación LED. A la hora de realizar la estimulación se hará uso de la información de la fase y la frecuencia y se recogerán las señales cerebrales con electrodos secos colocados en las posiciones mencionadas. El objetivo es mejorar el rendimiento del sistema, y por último se cuantificará la consecución de este objetivo midiendo las tasas de acierto y la usabilidad.

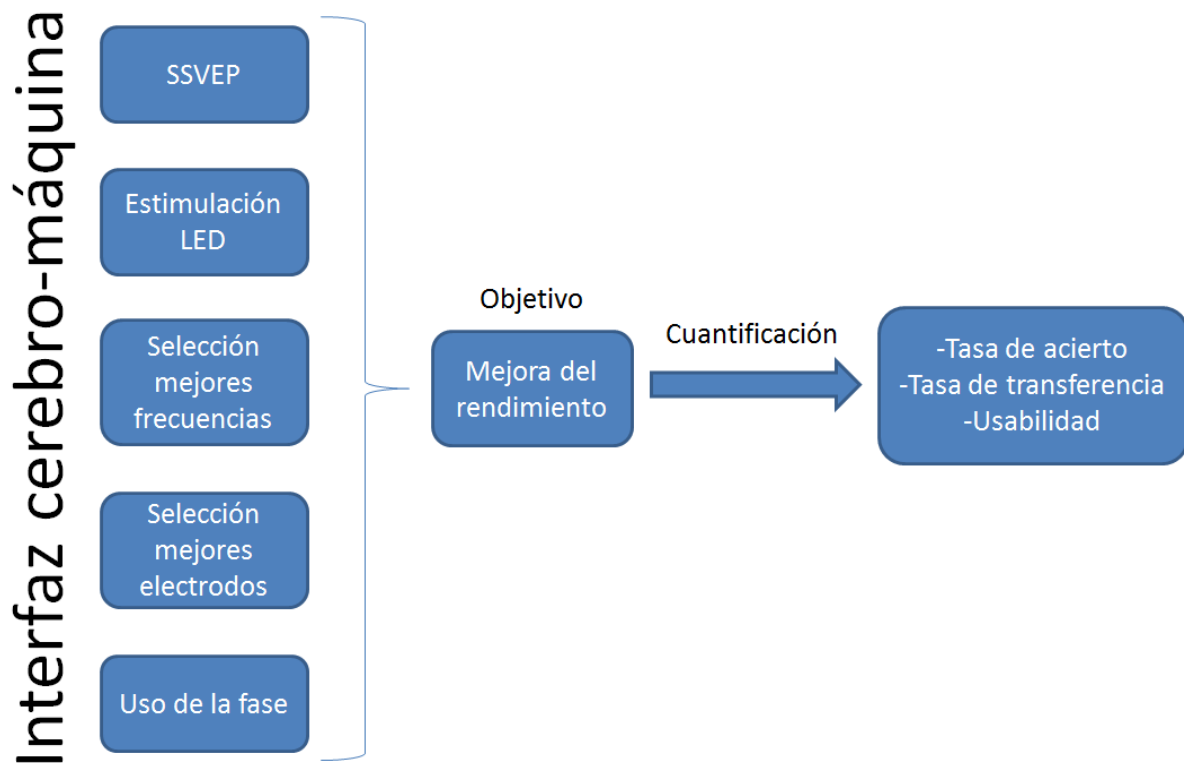


Ilustración 9: Esquema de la introducción de posibles mejoras de una interfaz basada en SSVEP tanto en el rendimiento como en la adaptación al usuario dentro del contexto de este proyecto.

3.6.1 Codificación de la fase

El primer paso a realizar en este proyecto es la codificación de la fase en el estímulo visual. Para ello se debe implementar el código de tal forma que el primer ciclo de la señal dure menos que un ciclo normal de la señal y así conseguir la fase deseada para cada estímulo.

Lo más sencillo sería usar la función `delay()` de Arduino, sin embargo, como se utilizarán 6 LEDs de forma simultánea, no se puede implementar de esta forma ya que al aplicar el retraso a uno de los LEDs también afectaría al resto, siendo imposible generar el estímulo con la fase y la frecuencia que deseemos para todos los LEDs. Por tanto, se seguirá el método utilizado por Ana Sotomayor Romillo en su TFM “Control adaptado al sujeto de una interfaz cerebro-maquina con señalización mixta” para generar las distintas frecuencias. Con este método se codifica la señal a la frecuencia deseada del modo siguiente: En primer lugar, se calcula el período a partir de la frecuencia. Después se inicializa la variable con la que se mide el tiempo en el que el LED cambió de encendido a apagado la última vez o viceversa. Se mide el tiempo actual y se calcula cuánto tiempo ha pasado desde la última vez que el LED se apagó o se encendió. Si el tiempo transcurrido es mayor que el periodo deseado, el LED cambia de estado. Este método puede hacerse extensible para codificar la fase. Para ello inicializamos una variable para indicar si se trata del primer periodo de la señal. Si es así, dependiendo de la fase que se quiera codificar se implementa el código para que el primer periodo tenga una duración mayor o menor.

Para explicarlo mejor veamos un ejemplo. Para una frecuencia de 5 Hz, el periodo será de 0.2 segundos, es decir, el LED estará 0.1 segundos encendido y 0.1 segundos apagado. Si queremos que la señal de 5 Hz tenga una fase de $\pi/2$, el primer tramo en el que el LED esté encendido deberá tener una duración de 0.05 segundos como se muestra en la siguiente ilustración.

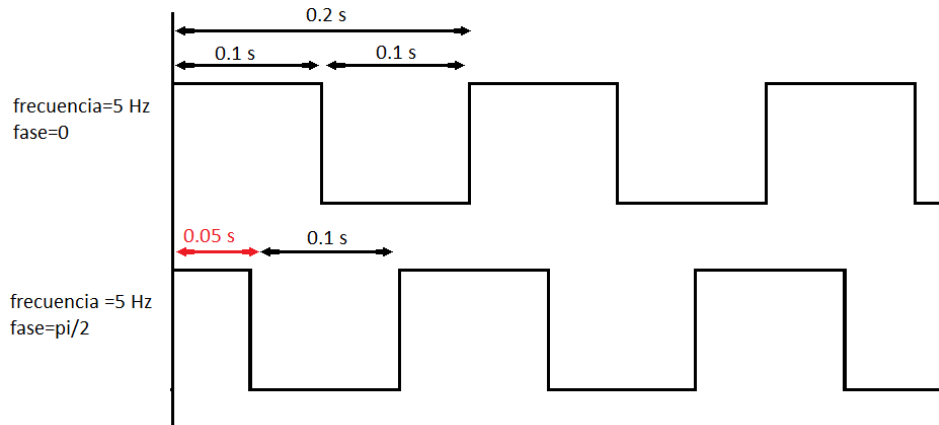


Ilustración 10: Ejemplo señales de los LED para frecuencia de 5Hz y fases 0 y $\pi/2$.

El código implementado para codificar la fase es el siguiente:

```
void EncenderLeds(unsigned long interval1)
{
    //Se obtiene para cada pasada de bucle el tiempo actual en microsegundos
    currentMicros1 = micros();

    //Codigo LED1
    // Si la diferencia entre el tiempo actual (currentMicros) y la última vez que el LED
    // parpadeo (previousMicros) es mayor que el periodo de oscilación, el LED cambia de
    // estado (Si está a HIGH pasa a LOW y viceversa).

    if (interval1==0)
        digitalWrite(led1Pin, LOW);
    else if(contador_fase==0)
    { //La variable contador_fase indica que se trata del primer periodo. Para fase = pi/2:
      led1State = 255;
      analogWrite(led1Pin, led1State);
      //Como se quiere usar fase pi/2 la variable interval1, que es la mitad del period se divide
      //entre 2
      if(currentMicros1 - previousMicros1 >= interval1/2){
          //Se guarda el microsegundo en el que cambio el estado
          previousMicros1 = currentMicros1;
          led1State = LOW;
          analogWrite(led1Pin, led1State);
      }
    }
```



```

        //Se actualiza la variable contador_fase a 1 para indicar que ya ha pasado el
        //primer periodo y ya se ha codificado la fase
        contador_fase=1;
    }
}

else if(currentMicros1 - previousMicros1 >= interval1)
{
    //Se guarda el microsegundo en el que cambio el estado
    previousMicros1 = currentMicros1;

    if (led1State == LOW)
        led1State = 255;
    else
        led1State = LOW;

    // Se pasa al pin el valor del estado
    analogWrite(led1Pin, led1State);
}

```

Tras empezar a realizar pruebas en sujetos con este tipo de estimulación, se comprobó que era molesto para los usuarios. Por ello se implementó otro modo de generar los estímulos mediante una señal sinusoidal que no cambiara de forma tan brusca de encendido a apagado. De este modo, se generó la señal con la correspondiente codificación de la fase utilizando la función `sin()` de Arduino.

```

currentMicros1 = micros();
// Se obtiene el tiempo en microsegundos y se pasa a segundos
currentSegundos1=currentMicros1/1000000;
led1State=sin(currentSegundos1*2*pi*led1f+fase)*255/2+255/2;

```

Con este método se mejora la comodidad del usuario, algo que resulta beneficioso debido a que el usuario puede fijar mejor la atención en el estímulo.

3.6.2 Detección de frecuencia y fase

El método más usado para detectar la frecuencia de los SSVEPs en este tipo de interfaces ha sido normalmente el del análisis de la densidad espectral de potencia. Este método consiste en realizar el estudio en frecuencia de la señal registrada con la FFT (transformada rápida de Fourier). De este modo se encuentran las frecuencias en las que la señal tiene mayor amplitud y esta frecuencia se puede relacionar con la frecuencia del comando que el usuario quiere transmitir.

Sin embargo, estudios recientes (Nakanishi et al., 2015; Sengelmann et al., 2017) han demostrado que los sistemas que utilizan la detección basada en el análisis de correlación canónica (CCA) mejoran la tasa de transmisión.

Este método de CCA utiliza dos conjuntos de variables, el primer conjunto son las señales recogidas de los distintos canales del casco de EEG, mientras que el segundo conjunto se trata de las señales que generan el estímulo. Además, en el caso de que se utilice una señal cuadrada una señal cuadrada para generar los estímulos con los LEDs, esta se puede descomponer en un conjunto de señales sinusoidales a la misma frecuencia y sus armónicos como se puede ver en la siguiente ilustración.

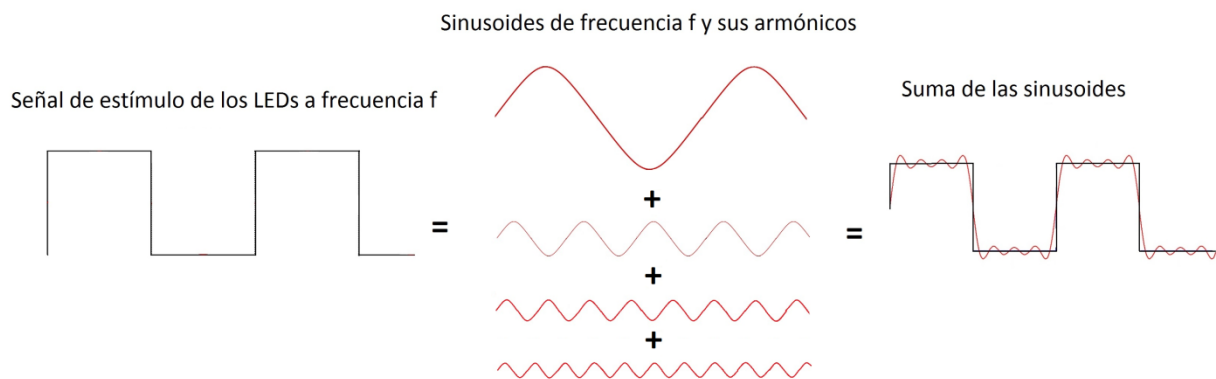


Ilustración 11: Descomposición de una señal cuadrada en señales sinusoidales.

Por tanto, el método de CCA se basa en la suposición de que las señales registradas en el EEG son la salida de un sistema lineal en el que la entrada son las señales de los estímulos, es decir, la descomposición en sinusoides de la señales de los LEDs. El análisis de correlación canónica encuentra los coeficientes que maximizan la correlación entre la entrada y la salida del sistema y de este modo, la frecuencia de estímulo con mayor coeficiente de correlación es la frecuencia detectada (Lin et al., 2007).

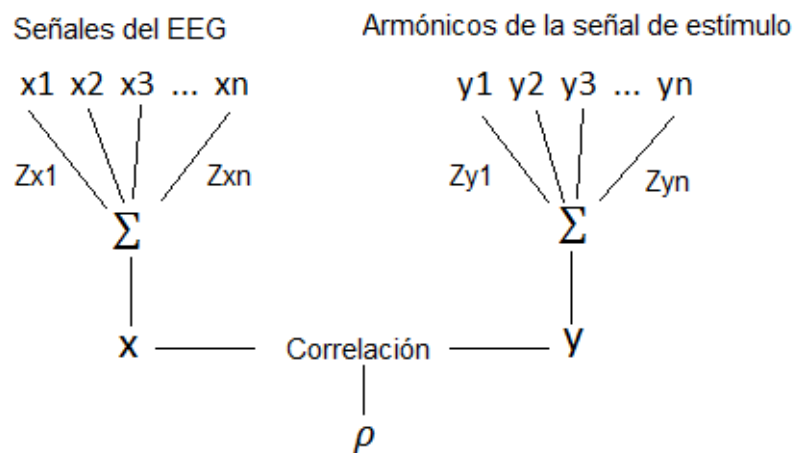


Ilustración 12: Explicación CCA. Adaptado de (Lin et al., 2007)

En la ilustración 12 se puede observar el funcionamiento del método de CCA. Las señales x_1, \dots, x_n son las señales de los distintos canales del EEG y las señales y_1, \dots, y_n son las señales y armónicos de la frecuencia de estímulo. El método de CCA encuentra los coeficientes Zx_1, \dots, Zx_n y Zy_1, \dots, Zyn que proporcionan la mayor correlación entre X e Y de modo que la frecuencia de estímulo f cumple que $f = \max(f)$ donde $\rho(f)$ es el coeficiente de correlación de X e Y.

4 Integración, pruebas y resultados

En este apartado se explicará el procedimiento realizado en las pruebas llevadas a cabo para comprobar el correcto funcionamiento del dispositivo, así como los resultados obtenidos.

4.1 Validación de frecuencias y fases

Con el objetivo de evaluar la precisión temporal de las señales de control del estímulo en cuanto a frecuencia y fase generadas con una placa Arduino se realizaron varias pruebas. Estas pruebas fueron llevadas a cabo con una tarjeta de adquisición NI PCI-6251 disponible en el laboratorio del grupo de Neurocomputación biológica de la EPS. Este sistema de adquisición cuenta con una resolución de entrada analógica de 16 bits con un rango de voltaje mínimo de -100mV a 100mV y una precisión de 52 μ V. Se utilizó una frecuencia de muestreo de 10KHz, es decir, una frecuencia muy superior a la de las frecuencias de estimulación del sistema BCI (hasta 22 Hz). Para llevar a cabo el experimento se programó el dispositivo Arduino para generar señales a diferentes frecuencias y fases y de este modo comprobar que la precisión en la generación del estímulo a distintas frecuencias y fases era la deseada.

Las frecuencias seleccionadas fueron 7 Hz, 10 Hz, 15Hz y 22Hz, cada una de ellas con 4 fases diferentes, fases igual a 0, $\pi/2$, π y $3\pi/2$. Se seleccionaron estas frecuencias y fases para simular un entorno real de uso de la interfaz cerebro máquina, es decir, una interfaz con 4 estímulos para que el usuario elija una opción de esos 4 estímulos, cada uno con una fase y una frecuencia diferente. Solo se realizó la prueba para una frecuencia menor de 10 Hz debido a que las frecuencias menores de 10 Hz suelen resultar molestas para los usuarios. Además, se tuvo en cuenta que ninguna de las frecuencias de estímulo utilizadas en el experimento son múltiplos entre sí, lo que en un caso real de uso facilita la decodificación de la señal que se obtiene del EEG.

Las medidas realizadas se llevaron a cabo con una frecuencia de muestreo de 10 KHz y una duración de cada prueba de 50 segundos. Se realizaron pruebas de esa duración para comprobar que la incorporación de la fase en la generación del estímulo no incluía problemas ni errores en la frecuencia del estímulo.

Los resultados son los mostrados en las siguientes ilustraciones, en las que se pueden ver las fases iniciales al comenzar a generar el estímulo, los espectros de potencia obtenidos a partir de las FFT realizadas en Matlab y las variaciones en el tiempo entre subidas de la señal.

En la ilustración 13 se pueden observar los momentos iniciales de las 4 señales que generan el estímulo en el Arduino para la interfaz cerebro-máquina. La primera grafica muestra la señal de 7 Hz con fase 0, la segunda, frecuencia de 10 Hz y fase $\pi/2$, por lo que el primer ciclo de encendido de la señal dura la mitad de lo que debería, la tercera, frecuencia 15 Hz y fase π , por lo que comienza con un ciclo apagado y la cuarta, frecuencia 22 Hz y fase $3\pi/2$, por lo que comienza con las mitad de un ciclo apagado.

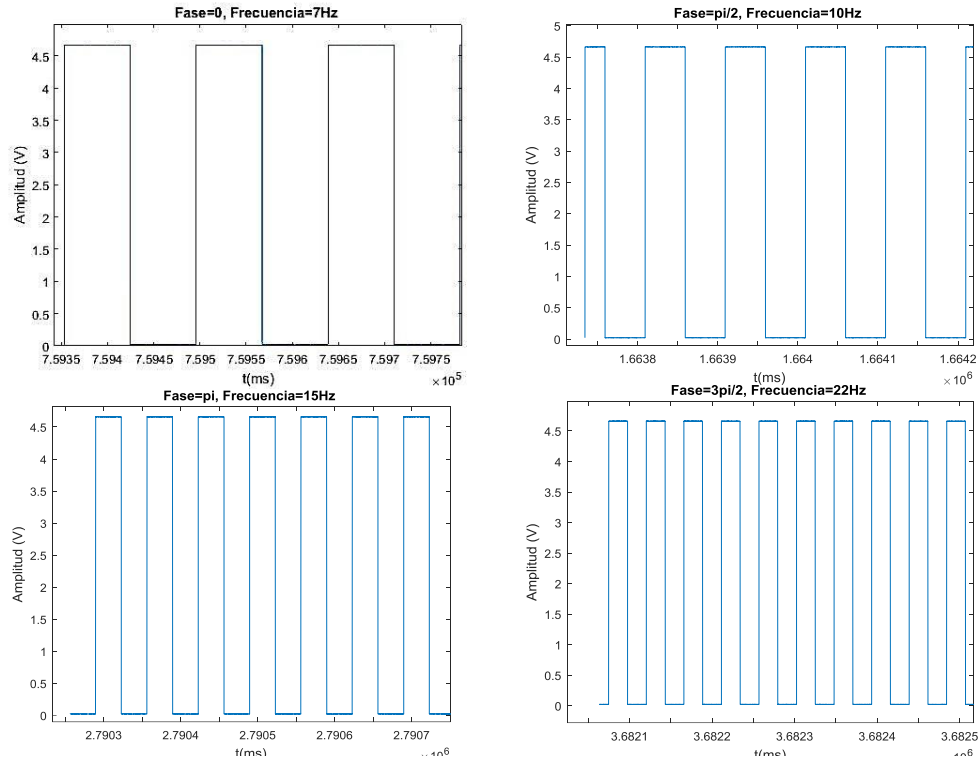


Ilustración 13: Ejemplo de señales de estímulo generadas con distintas fases (0 , $\pi/2$, π , y $3\pi/2$) y frecuencias (7, 10, 15 y 22 Hz).

Como se puede ver en la ilustración 14 las frecuencias obtenidas al realizar la FFT se corresponden con las frecuencias que se deseaban generar con un error máximo de 0.02 Hz.

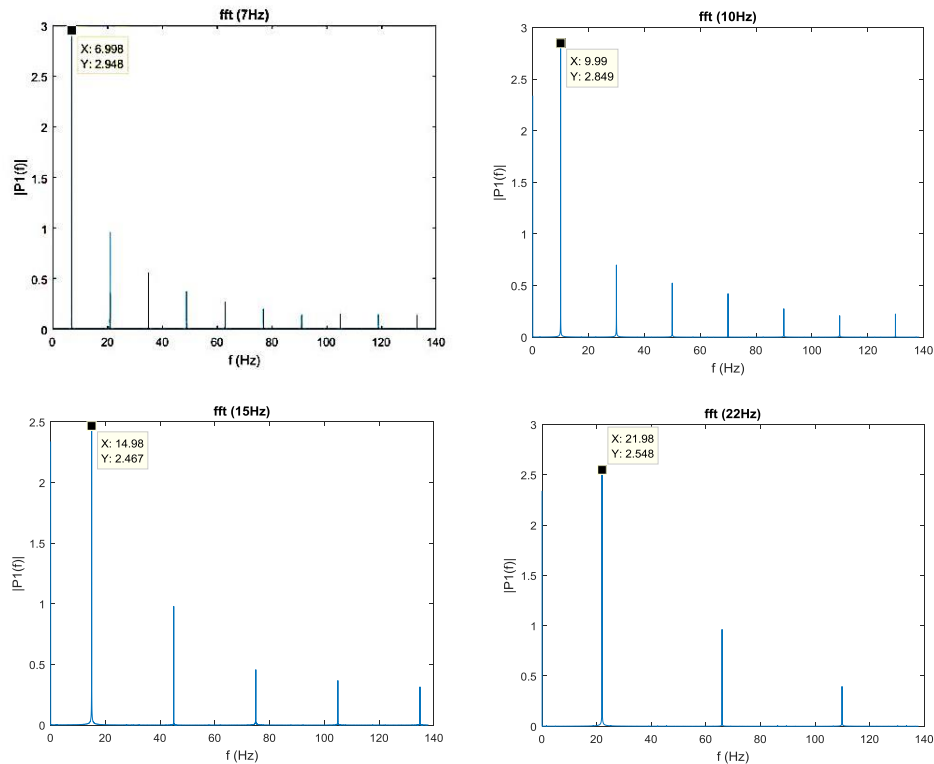


Ilustración 14: FFT de las señales de generación de los estímulos.

Para evaluar la precisión temporal de la generación del estímulo se midió ΔT como la diferencia de tiempos entre subidas de las señales y se representó esta diferencia en la ilustración 15. Se puede observar en esta figura que los periodos no son estrictamente regulares y por tanto hay un *jitter* temporal. En la ilustración se muestran la diferencia de tiempo entre subidas. En el caso de la señales a 10, 15 y 22 Hz también se muestra un zoom de las medidas de la diferencia de tiempo entre subidas eliminando el primer periodo, ya que al tener fases $\pi/2$, π , y $3\pi/2$ respectivamente el primer periodo dura menos y en las gráficas no se distingue el *jitter* temporal que si se distingue en la gráfica de la señal de 7 Hz y fase 0.

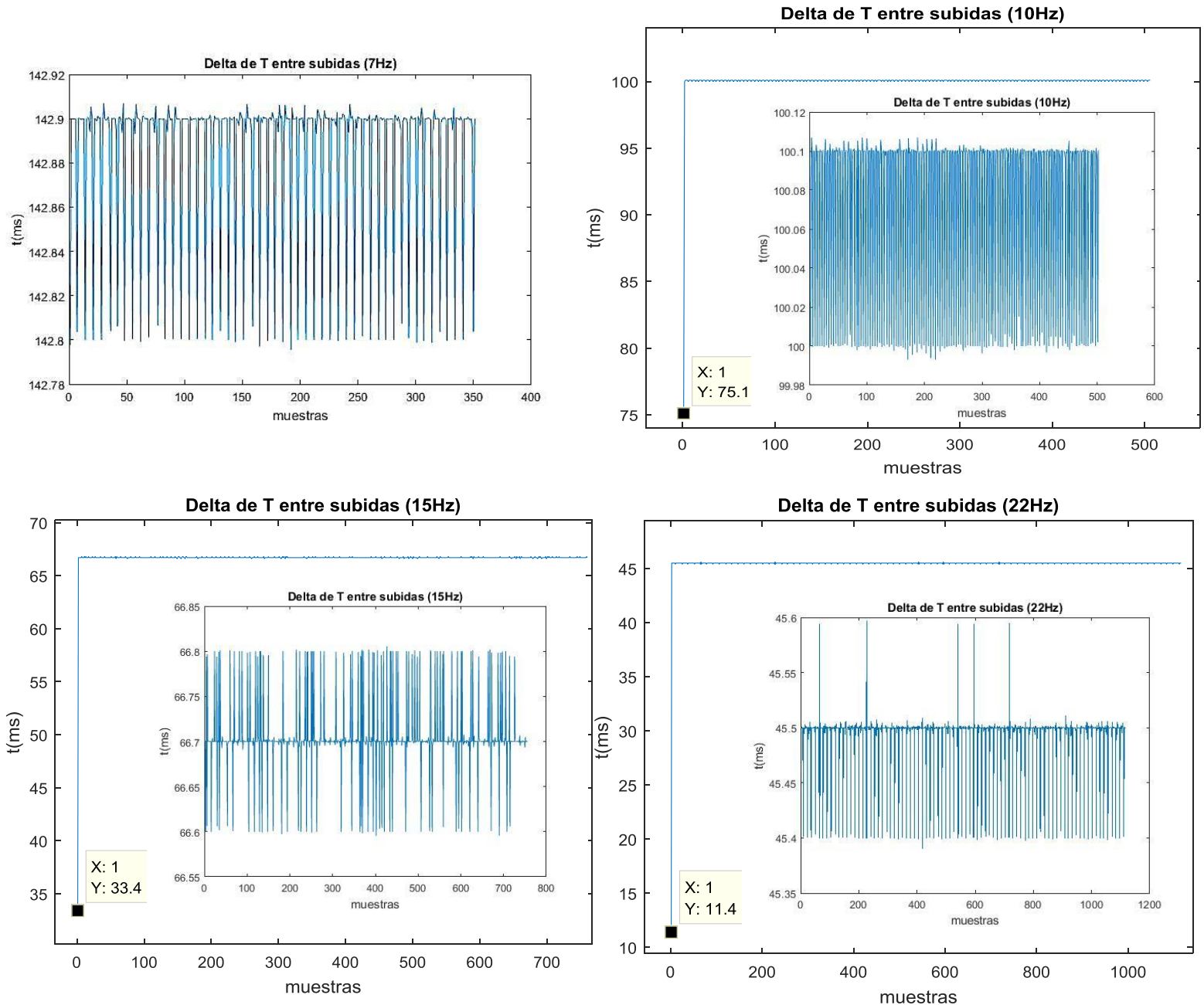


Ilustración 15: Evaluación del *jitter* temporal en la generación de señales para distintas frecuencias y fases.

Para el caso de 7 Hz, en la ilustración 15 se puede ver el tiempo entre subidas para los 350 periodos que se corresponden con los 50 segundos de medición. El periodo de una

señal de 7 Hz es de 142.8571429 ms y vemos que las medidas obtenidas tienen un periodo que oscila entre 142.8 y 142.95 ms, es decir el *jitter* es del orden de décimas de milisegundo. Para el caso de 10 Hz y fase $\pi/2$, el periodo es de 100 ms, por tanto, el primer periodo de la señal debería tener un valor de 75 ms y se obtiene un valor de 75.1 ms en la medida. El resto de periodos oscilan entre los 100 y los 100.1 ms. Para el caso de la señal a 15 Hz y fase π , el periodo es de 66.6666 ms, por tanto, el primer periodo debería valer 33.3333 ms y en la medida se obtiene un valor de 33.4 ms. El resto de periodos oscilan entre 66.8 y 66.6 ms. Para el último caso en el que la señal es de 22 Hz y fase $3\pi/2$, el periodo es de 45.4545 ms, por lo que el primer periodo de la señal debería valer 11.3636 y en la medida se obtiene un valor de 11.4 ms. La duración del resto de periodos oscila entre 45.4 y 45.6 ms.

Con estos resultados obtenidos en la primera prueba se concluye que el error obtenido en la generación de la señal es del orden de décimas de milisegundos para el rango de frecuencias de estimulación, un nivel de error que no influye de forma significativa en la generación de la frecuencia deseada, teniendo en cuenta que un registro típico de EEG se hace a una frecuencia inferior a 1Khz.

4.2 Prueba fotodiodo

La segunda prueba realizada se llevó a cabo midiendo la temporalidad de la luz emitida por los LEDs de la interfaz cerebro-máquina mediante un fotodiodo. Para ello se utilizó la misma tarjeta de adquisición que en la prueba anterior, disponible en el laboratorio del grupo de Neurocomputación biológica. El objetivo de esta prueba fue el de constatar que no hay retardo entre la generación de la señal por el dispositivo Arduino y la emisión de luz en los LEDs de la interfaz.

Los circuitos utilizados para llevar a cabo la medición con los fotodiodos fueron fabricados para esta prueba con un fotodiodo, una resistencia de 10 Ω y un conector BNC, procediéndose a su soldadura de modo que se obtuviera el siguiente circuito:

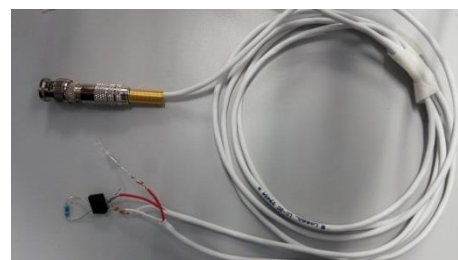
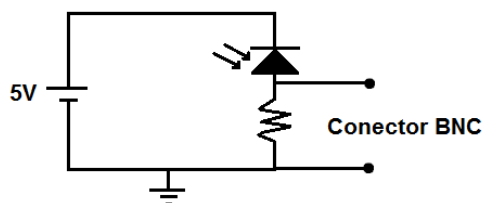


Ilustración 16: Circuito del fotodiodo y conector BNC y foto del dispositivo construido..

Para ello se realizaron las medidas simulando un ejemplo de utilización real de la interfaz, es decir, usando en este caso 6 frecuencias con 6 fases distintas, una para cada LED. En primer lugar se realizó la medida implementando el código de Arduino para manejar un solo LED individualmente. Las 6 frecuencias utilizadas fueron 7, 10, 13, 15, 17 y 22 Hz, y las fases utilizadas, 0, $\pi/3$, $2\pi/3$, π , $4\pi/3$ y $5\pi/3$ respectivamente. Los resultados se muestran en las siguientes gráficas. En la ilustración 17 se puede apreciar como la luz

emitida por los LEDs y recogida por el fotodiodo sigue manteniendo la información de la fase obtenida en la generación de la señal.

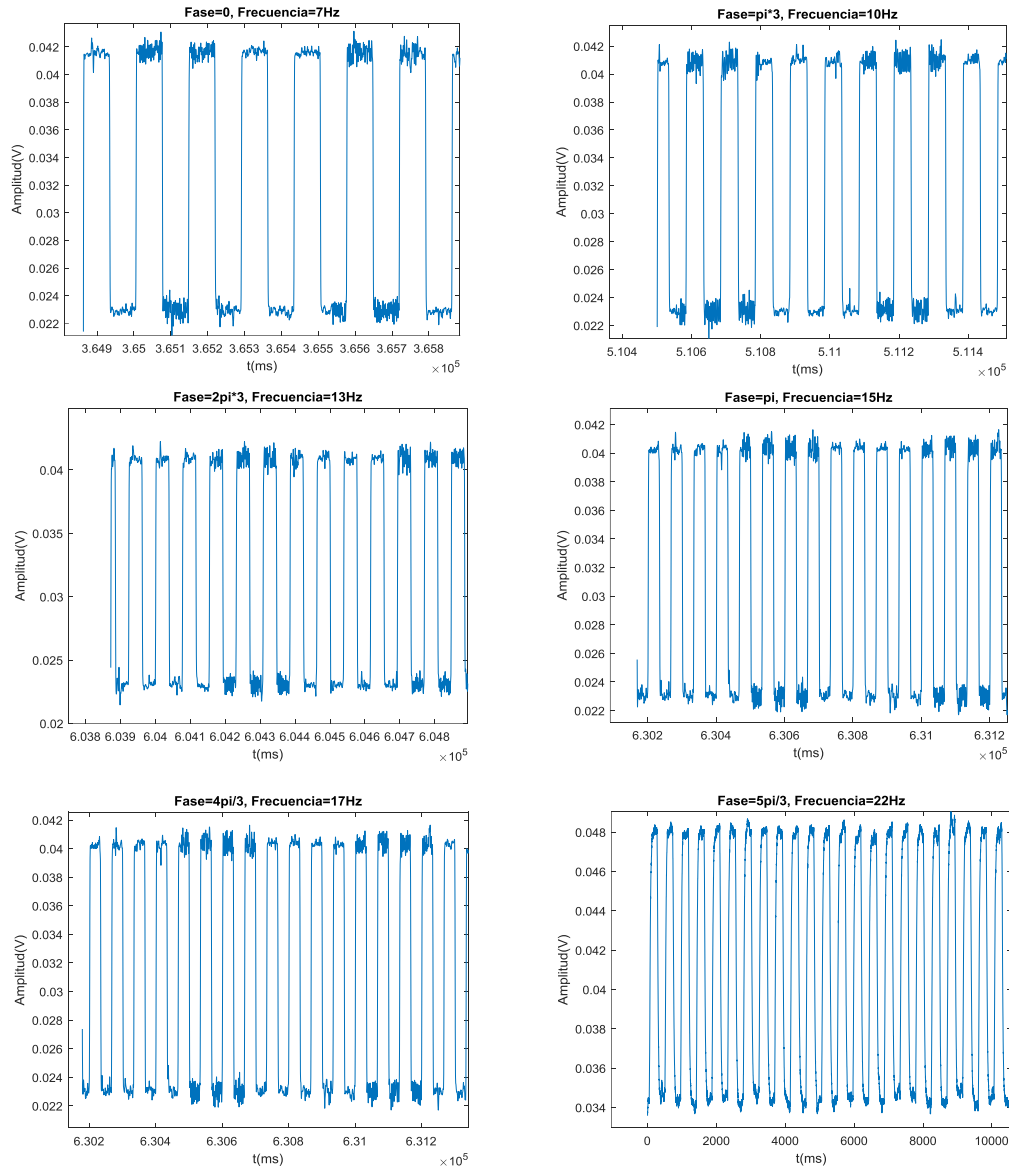


Ilustración 17: Medidas del fotodiodo durante la generación de comandos de encendido para un solo LED a la vez.

Tras esto se realizó la medición con todo el código de todos los LEDs parpadeando simultáneamente, es decir con la placa Arduino manejando los 6 LEDs a la vez para comprobar que la utilización de 6 LEDs con frecuencia y fase distintas es asumible para la placa de Arduino y no se producen retardos a la hora de encenderlos. Se recuerda que en un interfaz cerebro-máquina basado en SSVEP todos los LEDs tienen que estar encendidos para que el usuario fije su atención visual en uno de ellos cada vez que quiera evocar el comando que codifica. Para ello se realizaron pruebas midiendo con solo dos fotodiodos a la vez por falta de equipo para alimentar 6 de ellos simultáneamente (a pesar de no medir la señalización de los 6 a la vez, el código de Arduino que estaba corriendo en la placa generaba los comandos para controlar los 6 LEDs). Los resultados se describen a

continuación, comparando los LEDs que tenían un desfase de π radianes entre sí para una visualización más clara.

En la ilustración 18 se compara las medidas obtenidas del LED 1 a 7 Hz y con fase 0 y el LED 4 a 15 Hz y con fase π . Se puede apreciar claramente el desfase de π radianes entre las dos señales. Como vemos, el LED 1 comienza con un ciclo emitiendo luz mientras que el LED 4 comienza con un ciclo sin emitir luz.

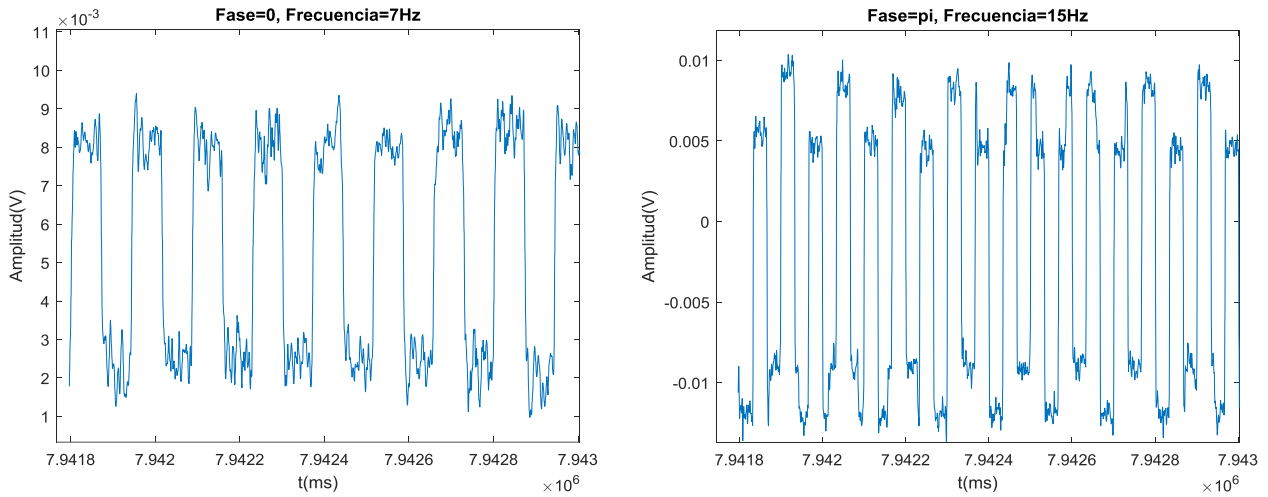


Ilustración 18: Medidas fotodiodo para LEDs 1 y 4.

En la ilustración 19 se compara las medidas obtenidas del LED 1 a 10 Hz y con fase $\pi/3$ y el LED 5 a 17 Hz y con fase $4\pi/3$. El LED 2 comienza con dos tercios del ciclo emitiendo luz mientras que el LED 4 comienza con dos tercios ciclo sin emitir luz.

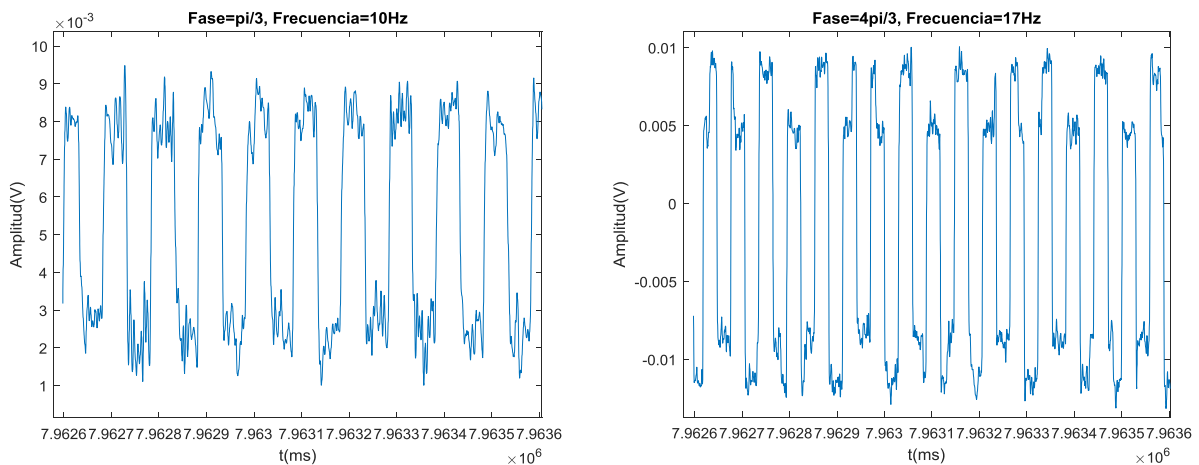


Ilustración 19: Medidas fotodiodo para LEDs 2 y 5.

En la ilustración 20 se compara las medidas obtenidas del LED 3 a 13 Hz y con fase $2\pi/3$ y el LED 6 a 22 Hz y con fase $5\pi/3$. El LED 3 comienza con un tercio del ciclo emitiendo luz mientras que el LED 4 comienza con un tercio del ciclo sin emitir luz.

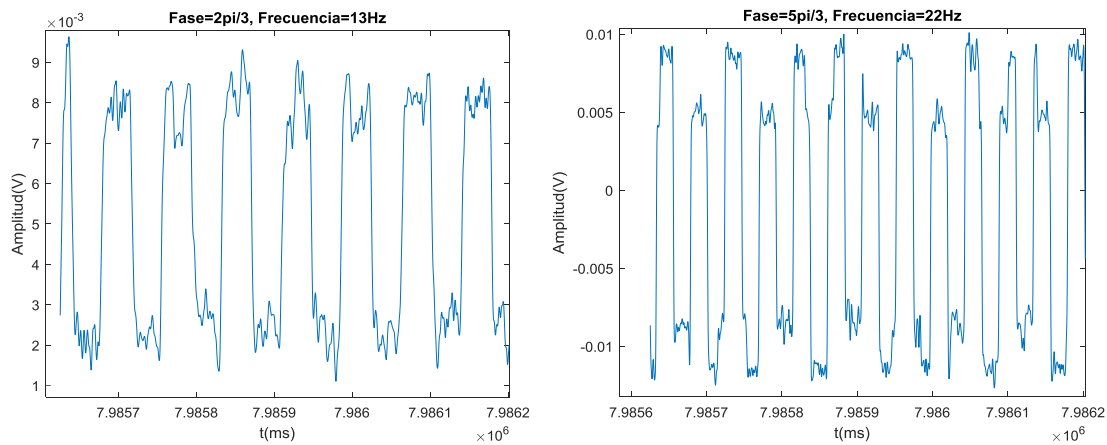


Ilustración 20: Medidas fotodiado para LEDs 3 y 6.

En la ilustración 21 se representan las señales medidas del fotodiado junto con las señales generadas en la placa Arduino. Se puede observar cómo la luz emitida por los LEDs se corresponde con la temporalidad de la señal generada. Los valores de las señales medidas con los fotodiodos (mostradas en rojo en la ilustración 21) se escalaron multiplicándolos por un factor de 100 para mejorar su visualización.

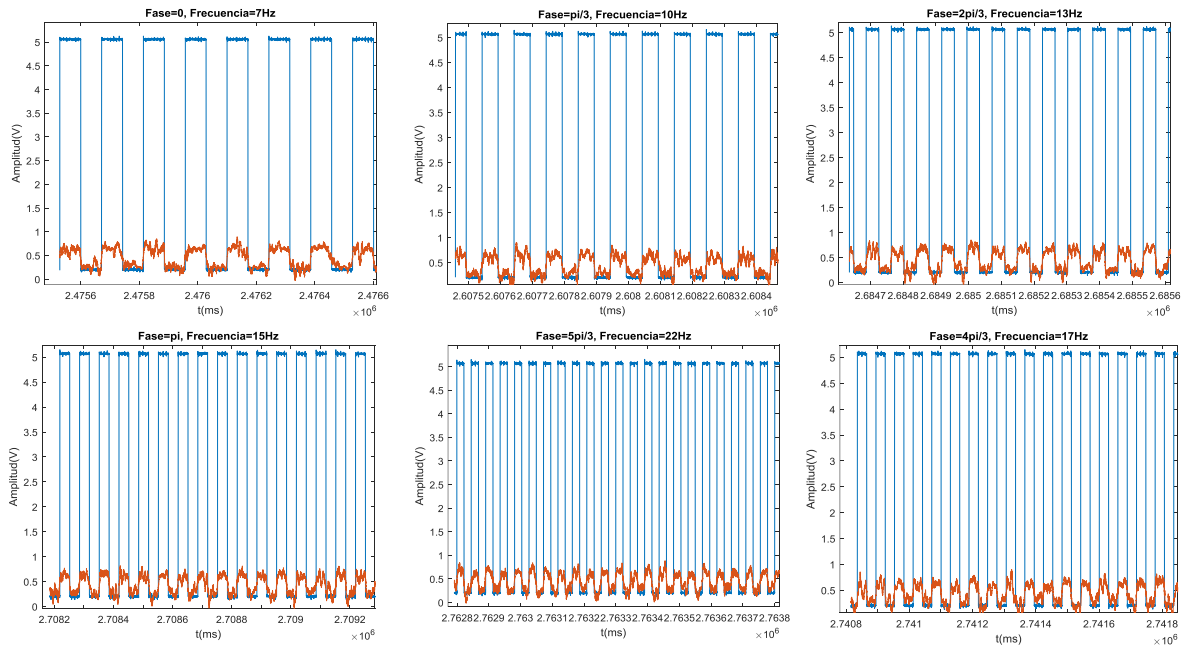


Ilustración 21: Medidas LEDs junto a medidas fotodiodos.

En la ilustración 22 se pueden observar el tiempo entre encendido de la luz emitida por el LED y en la tabla 1 se muestra el error de tiempo de encendido para el primer periodo de la señal, en algunos casos superiores a 1ms. Se observa que los resultados,

como era de esperar, son algo peores que para el caso en el que se midió la señal generada por Arduino. En este caso para el LED de 7Hz y fase 0, se obtienen unos periodos en el intervalo entre 134 y 152 ms, cuando se está generando un periodo de 142 ms.

Para el segundo LED a 10 Hz y fase $\pi/3$, el primer periodo dura 84.1ms cuando debería durar 83.3ms. Con un error de 1.1 ms. En el caso del tercer LED con frecuencia 13Hz y fase $2\pi/3$, el primer periodo tiene una duración de 52.1ms en vez de 51.28ms obteniendo un error de 0.9ms. El cuarto LED a frecuencia 15Hz y fase π , tiene un error de 1.4 ms. EL quinto LED con frecuencia de 17 Hz y fase $4\pi/3$ tiene un error de 2 ms y 0.5 Hz. Por último, el LED de 22Hz de frecuencia y $5\pi/3$ de fase tiene un error de 1.02ms.

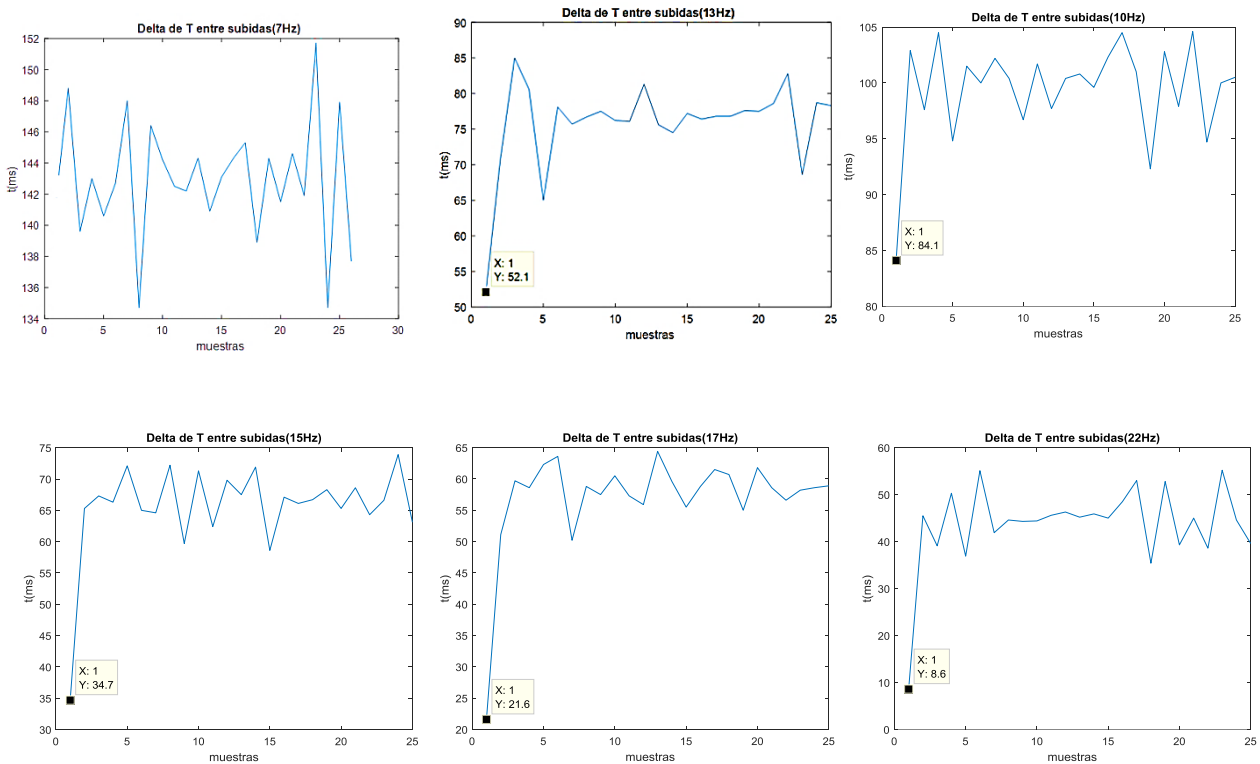


Ilustración 22: Medidas diferencias de tiempo entre subidas de la señal obtenida del fotodiodo.

	Fase 0 Frec=7Hz	Fase= $\pi/3$ Frec=10Hz	Fase= $2\pi/3$ Frec=13Hz	Fase= π Frec=15Hz	Fase= $4\pi/3$ Frec=17Hz	Fase= $5\pi/3$ Frec=22Hz
Duración teórica primer periodo	142.8ms	83.3ms	51.2ms	33.3ms	19.6ms	7.58ms
Duración obtenida primer periodo	143ms	84.1ms	52.1ms	34.7ms	21.6ms	8.6ms
Error	0.2ms	1.1ms	0.9ms	1.4 ms	2ms	1.02ms

Tabla 1: Duraciones primer periodo de la señal medida por el fotodiodo.

En la ilustración 23 se pueden observar los tiempos de latencia de encendido desde que se genera la señal de encendido en la placa de Arduino hasta que se enciende el LED correspondiente. Las medidas se muestran en la tabla 2, y dieron retrasos medios de en torno a 3 ms.

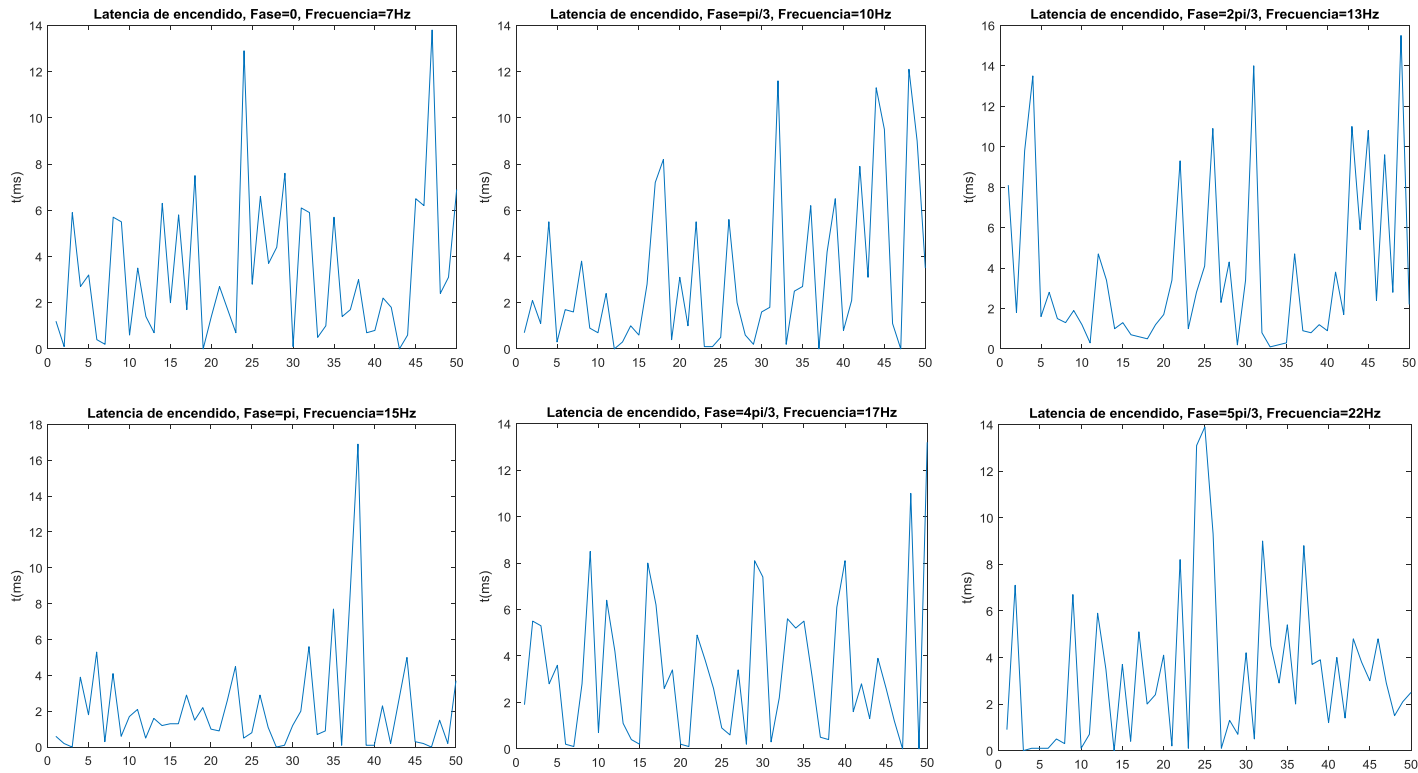


Ilustración 23: Medidas de latencia de encendido.

Latencia de encendido	Fase = π Frec = 7Hz	Fase = $\pi/3$ Frec = 10Hz	Fase = $2\pi/3$ Frec = 13Hz	Fase = π Frec = 15Hz	Fase = $4\pi/3$ Frec = 17Hz	Fase = $5\pi/3$ Frec = 22Hz
Media (ms)	3.3860	3.1540	3.8040	2.1400	3.4140	3.3500

Tabla 2: Medias de la latencia de encendido.

Para el caso particular de la estimulación con una forma de onda sinusoidal, como esta se generó a partir de una señal digital se realizó mediante PWM (pulse width modulation). Es decir, modificando el ciclo de encendido de los LEDs para que la intensidad de luz percibida se aproximara a una señal sinusoidal. Como se ha descrito anteriormente este método resultó menos molesto para los usuarios que una forma de onda cuadrada. Por tanto, un periodo de la señal generada, por ejemplo para el caso de 7 Hz, era el que se muestra en la ilustración 24. En ella se observa la señal generada por la placa Arduino, que va alternando ciclos de encendido y apagado de mayor o menor duración, de ese modo, al hacer la media del voltaje recibido por el LED se observa una señal sinusoidal.

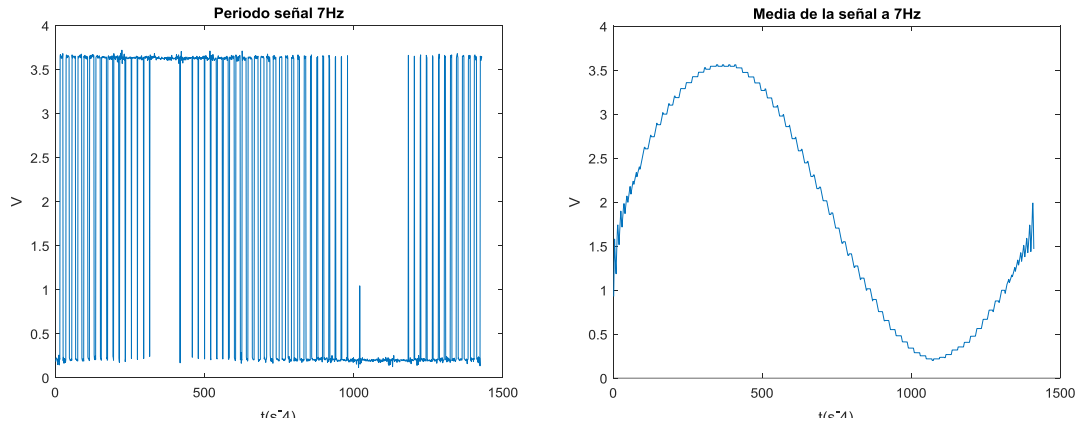


Ilustración 24: Periodo señal 7Hz PWM y su media.

Al generar la señal con una función seno, para la fase inicial 0, el valor inicial de la señal es de 0, lo que se corresponde con el LED a mitad de encendido, para la fase $\pi/2$, el valor inicial de la señal es de 1, que se corresponde con el LED totalmente encendido. Por último, el LED estará totalmente apagado en el instante inicial cuando tenga fase $3\pi/2$, como se puede ver en la siguiente ilustración, en la que se muestra el segundo inicial de la señal junto con la luz medida por el fotodiodo. En dicha ilustración, la señal medida por el fotodiodo (en rojo) se escaló multiplicándola por un factor de 300 para mejorar su visualización.

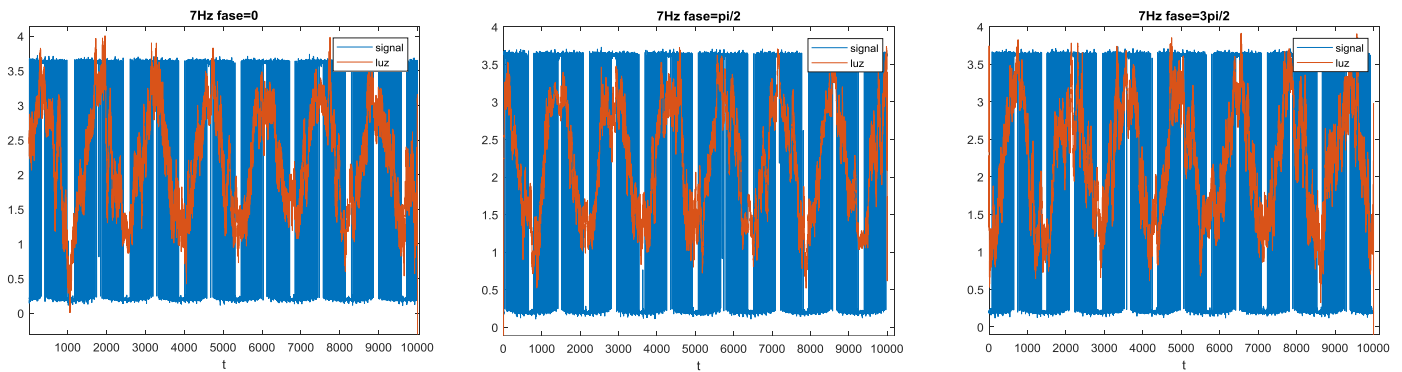


Ilustración 25: Señal sinusoidal generada con PWM y luz emitida.

Por tanto, tras la realización de las pruebas con el fotodiodo se concluye que las latencias de encendido máximas medidas de los LEDs que generaran las señales de estímulo están en torno a 3 ms, y por ello son aceptables para un interfaz cerebro-máquina que utilice frecuencias máximas del orden de 20 Hz como es el caso de este proyecto.

4.3 Pruebas en sujetos reales

Tras las pruebas realizadas a la señal generada y la luz emitida por los LEDs se llevaron a cabo las pruebas de detección de SSVEPs evocados por el parpadeo de los LEDs utilizando el sistema diseñado en este TFG con sujetos reales. Las pruebas se realizaron a 10 voluntarios que declararon previamente que no padecían de epilepsia o de molestias por iluminación parpadeante. Para ello se configuraron los dispositivos para realizar grabaciones de 7 segundos. De este modo se, pedía al usuario que fijara su atención en un LED, se grababa un segundo con el casco y amplificador de gTEC sin estimulación, es decir, con los LEDs apagados, y después se procedía a grabar durante 6 segundos mientras

ocurría la estimulación, con los LEDs parpadeando a una frecuencia y fase determinada. Las pruebas tenían una duración aproximada de media hora, por lo que se realizaron utilizando el método de estimulación con señal sinusoidal, para que resultara más cómodo a los usuarios.

Estas pruebas fueron llevadas a cabo conjuntamente con David Rodríguez Muñoz mientras realizaba su trabajo de fin de máster (Optimización y personalización de interfaces cerebro-máquina basados en potenciales visuales evocados).



Ilustración 26: Ejemplo pruebas de estimulación a sujetos reales con sistema de grabación gTEC.

La sincronización de la placa de Arduino y el conjunto de amplificador y casco de gTEC se realizó mediante un cable de salida y entrada digital. El programa de grabación registraba durante un segundo y después emitía una señal mediante el cable de salida digital que se hacía llegar a la placa de Arduino. La placa Arduino recibía esta señal que era tratada como una interrupción con la función `attachInterrupt()`, tras la cual se comenzaban a estimular con los LEDs. De este modo se consiguió la sincronización necesaria para, más tarde, poder discriminar señales utilizando la información de la fase. Por tanto, la señal obtenida tenía el siguiente aspecto mostrado en la ilustración 27, en la que se puede observar la señal obtenida de la diferencia entre los electrodos OZ y POZ (se ha realizado la diferencia para eliminar el ruido) junto con la señal de estimulación. Tras un segundo de grabación sin estímulo, comienza a estimularse y se puede apreciar la señal de SSVEP generada en la actividad cerebral a la misma frecuencia del estímulo.

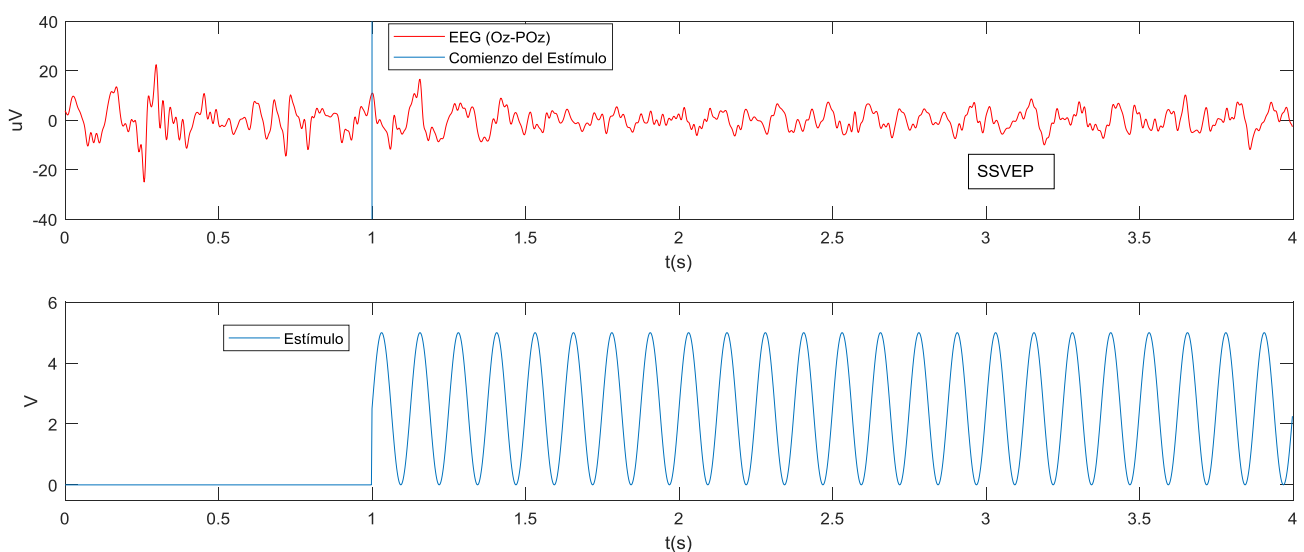


Ilustración 27: Ejemplo señal EEG con SSVEP v señal de estímulo.

Las grabaciones se realizaron estimulando a las frecuencias de 8, 9, 10, 11, 12, 13, 14 y 15 Hz con fases de 0 , $\pi/2$, π , $3\pi/2$, 0 , $\pi/2$, π y $3\pi/2$ respectivamente y 5 repeticiones de cada estímulo (realizando un total de 40 grabaciones). Se eligieron estas frecuencias y fases debido a que ninguna de las frecuencias es armónico de las demás y las fases son las más alejadas entre sí para el caso de una interfaz con 4 estímulos u opciones.

El método utilizado para la obtención del SSVEP a partir de las grabaciones de electroencefalograma fue el de correlación canónica (Rodríguez Muñoz, 2018). El procesamiento de las grabaciones se llevó a cabo de manera offline usando el programa MATLAB. El método de correlación canónica se llevó a cabo usando la función `canoncorr()`, pasándole como argumentos la señal de los 16 electrodos y dos señales sinusoidales, seno y coseno. Tras obtener las grabaciones se calculó cuáles fueron las frecuencias detectadas y para cada usuario se comprobó cual era la mejor frecuencia de estímulo seleccionando la frecuencia que obtenía los resultados con mayor amplitud. Las frecuencias obtenidas para cada usuario y la tasa de detección se muestran en la siguiente tabla:

	Número de detecciones	Tasa de detección	Mejor frecuencia
Sujeto 1	36	90%	10
Sujeto 2	39	97.5%	9
Sujeto 3	21	52.5%	10
Sujeto 4	40	100%	13
Sujeto 5	40	100%	9
Sujeto 6	39	97.5%	9
Sujeto 7	26	65%	8
Sujeto 8	21	52.5%	13
Sujeto 9	27	67.5%	9
Sujeto 10	38	95%	9

Tabla 3: Tasa de detección y mejores frecuencias para cada usuario.

Tras la obtención de la mejor frecuencia para cada usuario se procedió a realizar pruebas estimulando a cada usuario a la mejor frecuencia con distintas fases. Se realizaron 3 grabaciones para cada fase (0 , $\pi/2$, π y $3\pi/2$).

Para detectar la fase de los SSVEPs se realizó el siguiente procedimiento. Se tomaron las grabaciones del electrodo OZ con estímulo a partir de 140 ms después de que comenzara el estímulo, ya que son 140 ms lo que se estima de latencia visual entre que el sujeto empieza a recibir el estímulo y se produce una respuesta en su actividad cerebral (Chen et al., 2015). Filtrando esta señal a la frecuencia del estímulo se realizó la fft de dicha señal filtrada y se obtuvo su fase. Las fases obtenidas para cada sujeto se muestran en la ilustración 28. En esta ilustración se puede observar que para la mayoría de los sujetos se puede discriminar una señal de estímulo de otra a la misma frecuencia con fases distintas. De los 10 sujetos que realizaron la prueba solo hay 2 sujetos (sujetos 3 y 9) en los que no se podrían diferenciar las 4 fases. Para los 8 sujetos restantes, se observa como las señales generadas con la misma fase se agrupan en la gráfica de cada sujeto, diferenciándose del resto de señales de distinta fase, y por tanto, se podría discriminar en cuál de los LEDs está centrando su atención el sujeto. Los resultados se muestran en la ilustración 28.

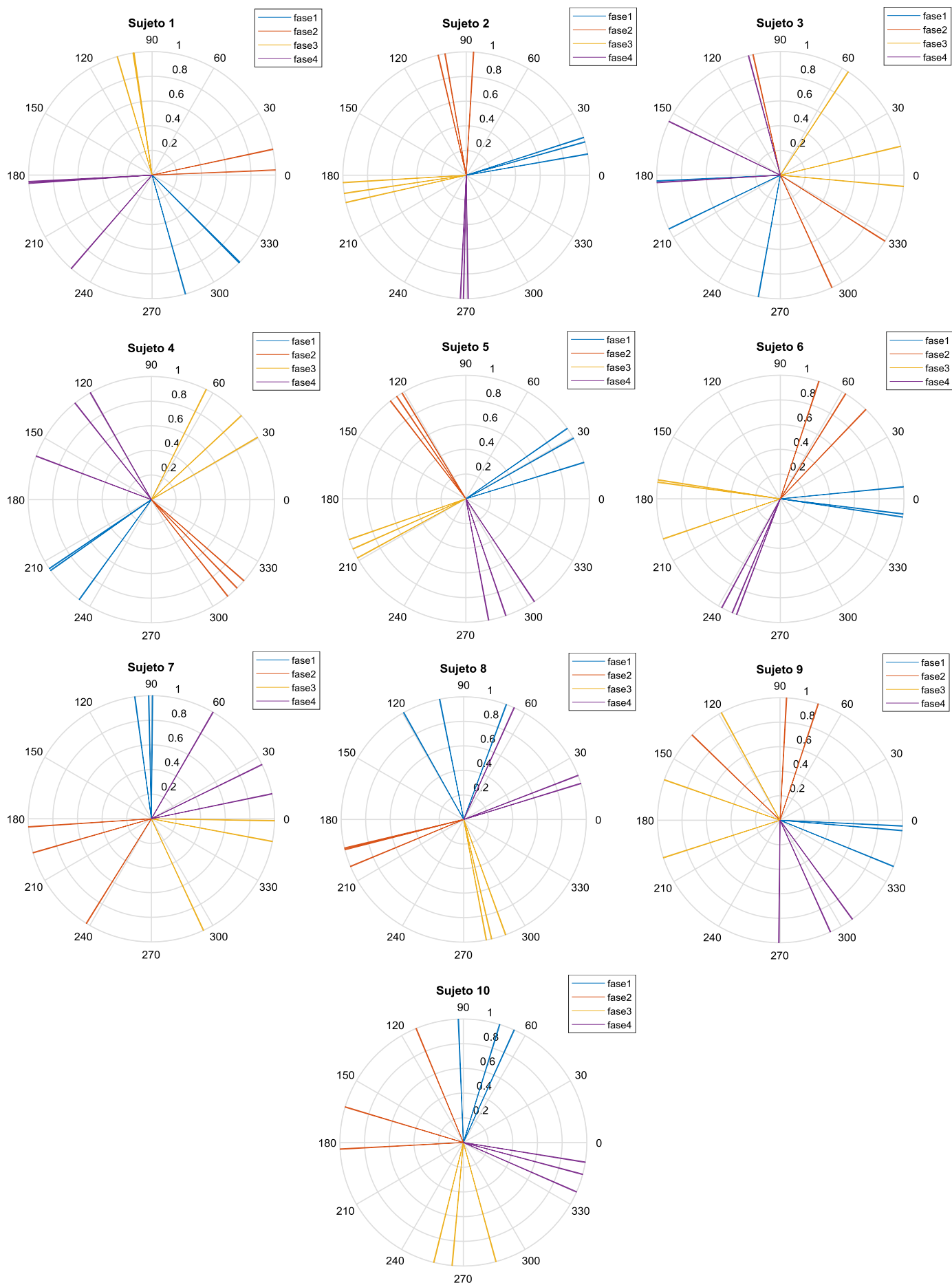


Ilustración 28: Resultados de las fases obtenidas.

Para entender mejor la figura 28 podemos centrarnos por ejemplo en el sujeto 5, que es el que obtuvo mejores resultados. En la gráfica del sujeto 5 se puede observar como las fases de las señales obtenidas del EEG al generar el estímulo con la fase 1 (0 radianes, es decir, 0 grados) se agrupan en torno a los 30 grados. Para el caso del estímulo generado con fase 2 ($\pi/2$ radianes, es decir, 90 grados), las fases de las señales detectadas se agrupan en torno a 120 grados. Para el estímulo generado con fase 3 (π radianes, es decir, 180 grados), las fases de las señales detectadas se agrupan en torno a 210 grados. En el último caso, para el estímulo generado con fase 4 ($3\pi/2$ radianes, es decir 270 grados) las fases de las señales detectadas se agrupan en torno a 300 grados. Con esto se concluye que aunque la fase de señal detectada no es la misma que la fase con la que se genera el estímulo (esto puede ser debido a que la latencia entre que se muestra el estímulo y se produce el SSVEP en la actividad cerebral varía para cada individuo) si es posible diferenciar una señal de otra utilizando la información de la fase. Los únicos sujetos para los que no se puede realizar una diferenciación de las 4 fases son los sujetos 3 y 9.

Por tanto se ha comprobado que el método de CCA es válido para la detección de SSVEPs generados con el dispositivo desarrollado en este trabajo, que el estímulo puede realizarse controlando la luminosidad de los LEDs siguiendo una forma de onda sinusoidal que resulta más cómoda para el usuario. Se ha comprobado también que mediante la adición de la información de la fase a la señal de estímulo se puede discriminar una señal de otra con distintas fases. Para un sistema BCI online se podría por tanto estudiar que frecuencias producen mejores resultados para el usuario y codificar varios comandos del BCI con la misma frecuencia pero distinta fase, lo que puede aumentar el número de opciones para este tipo de sistemas y a la vez aumentar su tasa de transmisión de información.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

En este trabajo de fin de grado se ha implementado un sistema para realizar la estimulación en una interfaz cerebro máquina mediante el uso del control de la frecuencia y la fase de LEDs que resulta más eficaz que el uso de estimulación con un monitor debido a que en este caso no se depende de la frecuencia de refresco de la pantalla. Adicionalmente, se ha implementado un método de generación de estímulo regulando el brillo de los LEDs siguiendo una forma sinusoidal, que ha resultado ser más cómoda para los usuarios. Asimismo, se ha comprobado que se puede controlar la frecuencia de los LEDs con un dispositivo Arduino y se han medido los retrasos temporales para controlar la información de la fase de la señal. Para ello se realizaron pruebas con la tarjeta de adquisición y con el sistema de grabación de electroencefalograma de gTEC disponibles en el laboratorio del Grupo de Neurocomputación Biológica.

Tras las pruebas de validación realizadas en este proyecto se han mostrado los retrasos temporales de la señal y con ellos se concluye que se puede utilizar la información de las fases en la detección de SSVEPs realizando la sincronización entre la grabación de EEG y la generación del estímulo visual. Además, se ha realizado la estimulación con LEDs de una forma que ha resultado ser más cómoda para los usuarios, evitando la señal cuadrada que provoca que el usuario se canse, pudiendo ser utilizado este sistema en otros setups de BCI. También se ha comprobado que es posible llevar a cabo la detección de SSVEPs mediante el método de correlación canónica, con tasas de acierto altas, alcanzando el 100% de acierto en varios usuarios. Por último, se ha comprobado que al generar señales de estímulo con la misma frecuencia y distinta fase se puede extraer la información de la fase de la señal grabada y discriminar si el usuario está centrando su atención en un estímulo u otro, de esta forma, si un usuario responde mejor a una frecuencia determinada, se pueden codificar varios estímulos con la misma frecuencia pero diferente fase, lo que se puede mejorar la detección de SSVEP utilizando información de frecuencia y fase. Estas mejoras pueden sacar del estado del *BCI illiteracy* a algunos usuarios.

5.2 Trabajo futuro

Como se ha mostrado en los resultados, algunos usuarios respondían mejor que otros a la detección de la fase, y además, las fases detectadas variaban entre individuos. Por tanto el primer paso a realizar en un trabajo futuro sería un sistema, que utilizando el método de detección de correlación canónica, además de comprobar qué frecuencias son las mejores para el usuario, compruebe también sus mejores fases, es decir que optimice la respuesta que tiene dicho usuario al estimularlo con fases también personalizadas y así poder codificar de la mejor forma los estímulos con las frecuencias y las fases que proporcionen mayor diferenciación entre los estímulos para ese usuario. Además se podría mejorar este sistema eligiendo los electrodos que proporcionen mejores resultados. Todo ello se debe integrar con una interfaz gráfica que proporcione las opciones a elegir para el usuario. Además se puede realizar un mejor método de filtrado de la grabación para eliminar ruido y mejorar la calidad de la señal.

Otra línea de trabajo futuro sería explorar algoritmos que utilizasen adicionalmente métodos de aprendizaje automático para optimizar detección de SSVEPs o hibridarla con otro tipo de señal para mejorar la tasa de transmisión de información. Sin embargo, el mayor problema de estos sistemas sigue siendo su aparatosidad y la incomodidad del casco, por lo que una mejora muy interesante sería incorporar las evoluciones de este tipo de tecnología en cuanto a que el casco fuera más ligero o más cómodo para el usuario, y los electrodos más precisos y con menos ruido.

Referencias

- Allison, B. Z., Wolpaw, E. W., and Wolpaw, J. R. (2007). Brain-computer interface systems: progress and prospects. *Expert Rev. Med. Devices* 4, 463–474. doi:10.1586/17434440.4.4.463.
- Carelli, L., Solca, F., Faini, A., Meriggi, P., Sangalli, D., Cipresso, P., et al. (2017). Brain-Computer Interface for Clinical Purposes: Cognitive Assessment and Rehabilitation. *Biomed Res. Int.* 2017. doi:10.1155/2017/1695290.
- Chan, A. T., Quiroz, J. C., Dascalu, S., and Frederick C. Harris, J. (2015). An Overview of Brain Computer Interfaces. Available at: <https://pdfs.semanticscholar.org/7423/c275d0393f52e11d96e377f4188e3df9ef19.pdf>.
- Chen, X., Wang, Y., Nakanishi, M., Gao, X., Jung, T.-P., and Gao, S. (2015). High-speed spelling with a noninvasive brain-computer interface. *Proc. Natl. Acad. Sci. U. S. A.* 112, E6058-67. doi:10.1073/pnas.1508080112.
- Choi, I., Rhiu, I., Lee, Y., Yun, M. H., and Nam, C. S. (2017). A systematic review of hybrid brain-computer interfaces: Taxonomy and usability perspectives. *PLoS One* 12, 1–35. doi:10.1371/journal.pone.0176674.
- Cossio, E. G., and Gentiletti, G. G. (2008). Interfaz cerebro computadora (ICC) basada en el potencial relacionado con eventos P300: análisis del efecto de la dimensión de la matriz de estimulación sobre su desempeño – Brain-computer interface based on the P300 event-related potential: analysis of t. *Rev. Ing. Biomédica* 2, 26–33.
- Duszyk, A., Bierzyńska, M., Radzikowska, Z., Milanowski, P., Kuś, R., Suffczyński, P., et al. (2014). Towards an Optimization of Stimulus Parameters for Brain-Computer Interfaces Based on Steady State Visual Evoked Potentials. *PLoS One* 9, e112099. doi:10.1371/journal.pone.0112099.
- Fernandez-Vargas, J., Pfaff, H. U., Rodriguez, F. B., and Varona, P. (2013a). Assisted closed-loop optimization of SSVEP-BCI efficiency. *Front. Neural Circuits* 7, Article 27. doi:10.3389/fncir.2013.00027.
- Fernandez-Vargas, J., Pfaff, H. U., Rodríguez, F. B., and Varona, P. (2013b). Assisted closed-loops for brain-computer interfaces. *BMC Neurosci.* 14, P406. doi:10.1186/1471-2202-14-s1-p406.
- Gunasekera, B., Saxena, T., Bellamkonda, R., and Karumbaiah, L. (2015). Intracortical Recording Interfaces: Current Challenges to Chronic Recording Function. *ACS Chem. Neurosci.* 6, 68–83. doi:10.1021/cn5002864.
- Herrmann, C. S. (2001). Human EEG responses to 1-100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Exp. Brain Res.* 137, 346–353. doi:10.1007/s002210100682.
- Hirschberg, J., and Manning, C. D. (2015). Advances in natural language processing. *Science* (80-.). 349, 261–266.
- Hong, K.-S., and Khan, M. J. (2017). Hybrid Brain-Computer Interface Techniques for Improved Classification Accuracy and Increased Number of Commands: A Review. *Front. Neurobot.* 11, 35. doi:10.3389/fnbot.2017.00035.
- Lebedev, M. (2014). Brain-machine interfaces: an overview. *Transl. Neurosci.* 5, 99–110. doi:10.2478/s13380-014-0212-z.
- Lebedev, M. A., and Nicolelis, M. A. L. (2006). Brain-machine interfaces: past, present and future. *Trends Neurosci.* 29, 536–546.
- Lin, Z., Zhang, C., Wu, W., and Gao, X. (2007). Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs. *IEEE Trans. Biomed. Eng.* 54, 1172–

1176. doi:10.1109/TBME.2006.889197.
- Maestú, F., González-Marqués, J., Marty, G., Nadal, M., Cela-Conde, C., and Ortiz, T. (2005). La magnetoencefalografía: una nueva herramienta para el estudio de los procesos cognitivos básicos. *Psicothema* 17, 459–464.
- Marcus, M., Baranes, K., Park, M., Choi, I. S., Kang, K., and Shefi, O. (2017). Interactions of Neurons with Physical Environments. *Adv. Healthc. Mater.* 6, 1700267. doi:10.1002/adhm.201700267.
- Nakanishi, M., Wang, Y., Wang, Y.-T., and Jung, T.-P. (2015). A Comparison Study of Canonical Correlation Analysis Based Methods for Detecting Steady-State Visual Evoked Potentials. *PLoS One* 10, 1–18. doi:10.1371/journal.pone.0140703.
- Nicolas-Alonso, L. F., and Gomez-Gil, J. (2012). Brain Computer Interfaces, a Review. *Sensors* 12, 1211–1279. doi:10.3390/s120201211.
- Ramos-Argüelles, F., Morales, G., Egozcue, S., Pabón, R. M., and Alonso, M. T. (2009). *Anales del sistema sanitario de Navarra*. Gobierno de Navarra, Departamento de Salud Available at: http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1137-66272009000600006&lng=es&nrm=iso&tlng=es [Accessed November 13, 2017].
- Rodríguez Muñoz, D. (2018). Optimización y personalización de interfaces cerebro-máquina basados en potenciales visuales evocados.
- Rozado, D., Agustin, J. S., Rodriguez, F. B., and Varona, P. (2012a). Gliding and saccadic gaze gesture recognition in real time. *ACM Trans. Interact. Intell. Syst.* 1, 10:1--10:27. doi:10.1145/2070719.2070723.
- Rozado, D., Moreno, T., San Agustin, J., Rodriguez, F. B., and Varona, P. (2015). Controlling a Smartphone Using Gaze Gestures as the Input Mechanism. *Human-Computer Interact.* 30, 34–63. doi:10.1080/07370024.2013.870385.
- Rozado, D., Rodriguez, F. B., and Varona, P. (2012b). Low cost remote gaze gesture recognition in real time. *Appl. Soft Comput.* 12, 2072–2084. doi:10.1016/j.asoc.2012.02.023.
- Sengelmann, M., Engel, A. K., and Maye, A. (2017). Maximizing Information Transfer in SSVEP-Based Brain-Computer Interfaces. *IEEE Trans. Biomed. Eng.* 64, 381–394. doi:10.1109/TBME.2016.2559527.
- Singla, R., Khosla, A., and Jha, R. (2013). Influence of stimuli color on steady-state visual evoked potentials based BCI wheelchair control. *J. Biomed. Sci. Eng.* 6, 1050–1055. doi:10.4236/jbise.2013.611131.
- Vargas, J. F. (2013). Optimization of SSVEP-BCIs through an assisted closed-loop.
- Wikipedia contributors (2018). 10-20 system (EEG) --- {Wikipedia}{,} The Free Encyclopedia. Available at: [https://en.wikipedia.org/w/index.php?title=10-20_system_\(EEG\)&oldid=829895818](https://en.wikipedia.org/w/index.php?title=10-20_system_(EEG)&oldid=829895818).
- Wittevrongel, B., Van Wolputte, E., and Van Hulle, M. M. (2017). Code-modulated visual evoked potentials using fast stimulus presentation and spatiotemporal beamformer decoding. *Sci. Rep.* 7. doi:10.1038/s41598-017-15373-x.
- Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clin. Neurophysiol.* 113, 767–791. doi:10.1016/S1388-2457(02)00057-3.
- Zephaniah, P. V., and Kim, J. G. (2014). Recent functional near infrared spectroscopy based brain computer interface systems: Developments, applications and challenges. *Biomed. Eng. Lett.* 4, 223–230. doi:10.1007/s13534-014-0156-9.
- Zhu, D., Bieger, J., Garcia Molina, G., and Aarts, R. M. (2010). A survey of stimulation methods used in SSVEP-based BCIs. *Comput. Intell. Neurosci.* 702357. doi:10.1155/2010/702357.

Glosario

Interfaz cerebro ordenador (BCI) o interfaz cerebro-máquina (BMI): Sistema de comunicación entre una persona y una máquina o un ordenador que tiene como objetivo controlar dicha máquina u ordenador mediante actividad cerebral.

Steady-State Visually Evoked Potentials (SSVEP): Señales cerebrales generadas en un individuo al recibir estímulos visuales parpadeando a una determinada frecuencia.

Fase: fracción del periodo transcurrido de una señal desde el instante tomado como referencia.

Arduino: plataforma de electrónica de hardware de fácil uso y software de código abierto. Este sistema está basado en una placa microcontroladora y un entorno de desarrollo software.

Electroencefalograma: grabación de la actividad cerebral mediante el uso de electrodos colocados en el cuero cabelludo que miden la diferencia de potencial eléctrico.

LED: diodo emisor de luz (light-emitting diode).

gTEC: empresa dedicada a la producción de hardware y software destinado a sistemas BCI.

PWM: siglas de “pulse width modulation”. Método para simular una señal analógica a partir de una señal digital, en la que el ancho del pulso va variando para obtener el voltaje deseado a transmitir.

Anexos

A Manual del programador

En este anexo se adjunta el código de Arduino utilizado en la generación de la señal:

```
//Inicializacion de variables globales
boolean ESTIMULACION=false;
boolean ESTIMULADO=false;
boolean PUERTO_SERIE_WAITING_MESSAGE=true;
boolean PUERTO_SERIE_CALLED=false;
const float pi=3.14159265359;

// Numero de puertos a los que se conectan los LEDs. Se seleccionan los pines PWM
const int led1Pin = 3;
const int led2Pin = 5;
const int led3Pin = 6;
const int led4Pin = 9;

// Estado inicial de los LEDs a LOW
int led1State = LOW;
int led2State = LOW;
int led3State = LOW;
int led4State = LOW;

//tiempo en microsegundos desde la última vez que se encendieron los LEDs
unsigned long previousMicros1 = 0;
unsigned long previousMicros2 = 0;
unsigned long previousMicros3 = 0;
unsigned long previousMicros4 = 0;

//frecuencias y fases de los LEDs y config_led
int trama_config = 0;
bool config_led;
float led1f;
float led2f;
float led3f;
float led4f;

float PHASE1;
float PHASE2;
float PHASE3;
float PHASE4;

//Tiempo actual en microsegundos
unsigned long currentMicros1 = 0;
unsigned long currentMicros2 = 0;
unsigned long currentMicros3 = 0;
unsigned long currentMicros4 = 0;

float resta1=0;
float resta2=0;
float resta3=0;
float resta4=0;

//Tiempo actual en segundos
float currentSegundos1=0;
float currentSegundos2=0;
float currentSegundos3=0;
float currentSegundos4=0;
|
void setup() {
  // Se declaran los pines como salidas
  pinMode(led1Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
  pinMode(led3Pin, OUTPUT);
  pinMode(led4Pin, OUTPUT);
  Serial.begin(9600);

  // Se declaran el pin por el que se recibe la interrupcion
  pinMode(2, INPUT);

  //Se trata la interrupcion para que llame a la funcion estimular
  attachInterrupt(digitalPinToInterrupt(2), Estimular, RISING);
}
}
```



```

void loop() {
    //Cuando haya que estimular se llama a la funcion EncenderLeds
    if (ESTIMULACION==!PUERTO_SERIE_WAITING_MESSAGE) {
        EncenderLeds();
    }
}

void serialEvent() {
    // Se tratan los eventos del puerto serie
    int var=0;
    if (!PUERTO_SERIE_CALLED){
        if (Serial.available()>1) {
            var=Serial.parseInt();
            //Si se recibe 179 por el puerto serie se apagan los LEDS
            if (var==179){
                Serial.println("Se ha llamado al puerto serie. Leds apagados, esperando a que se envíen las frecuencias y fases");
                Apagar();

                PUERTO_SERIE_CALLED=true;
                PUERTO_SERIE_WAITING_MESSAGE=true;
            }
        }
    }
    else {
        //Se reciben las frecuencias y fases por el puerto serie
        if (Serial.available()>1) {
            led1f=Serial.parseFloat();
            PHASE1=Serial.parseFloat();

            led2f=Serial.parseFloat();
            PHASE2=Serial.parseFloat();

            led3f=Serial.parseFloat();
            PHASE3=Serial.parseFloat();

            led4f=Serial.parseFloat();
            PHASE4=Serial.parseFloat();

            PUERTO_SERIE_WAITING_MESSAGE=false;
            PUERTO_SERIE_CALLED=false;
        }
    }
}

void Estimular(){
    if (!ESTIMULADO){
        //Se guarda el tiempo en micros en el que comienza la estimulación
        previousMicros1 = micros();
        previousMicros2 = micros();
        previousMicros3 = micros();
        previousMicros4 = micros();

        ESTIMULACION=true;
        ESTIMULADO=true;
    }
}

void Apagar(){
    // Se apagan los LEDS
    led1State = 0;
    analogWrite(led1Pin, led1State);
    led2State = 0;
    analogWrite(led2Pin, led2State);
    led3State = 0;
    analogWrite(led3Pin, led3State);
    led4State = 0;
    analogWrite(led4Pin, led4State);

    ESTIMULADO=false;
    ESTIMULACION=false;
}

void EncenderLeds()
{
    //Codigo LED1
    // Se obtien el tiempo actual y se calcula el tiempo desde que se comenzo a estimular
    //para calcular el estado de iluminacion del LED
    currentMicros1 = micros();
    resta1 =currentMicros1 - previousMicros1;
    currentSegundos1=resta1/1000000;
    led1State=sin(2*pi*led1f*currentSegundos1+PHASE1*pi)*255/2+255/2;
    //Serial.println(led1State);
    analogWrite(led1Pin, led1State);
}
//Este procedimiento se repetirá para los 6 LEDS

```

